

5        **SEMICONDUCTOR WITH SELECTIVE INTEGRATED  
CASHLESS, CONNECTIVITY, AND DIGITAL CONTENT  
PRESENTATION CAPABILITIES**

## RELATED APPLICATIONS

10

15 This U.S. non-provisional application claims priority of a U.S. provisional application, Serial No. 60/278,599 inventor H. Brock Kolls, entitled VENDING MACHINE AUDIT AND CREDIT CONTROLLER, SYSTEM AND METHOD, filed March 26, 2001; in addition this non-provisional application claims priority of a U.S. non-provisional application, Serial No. 10/051594, inventor H. Brock Kolls, entitled A WIRELESS SYSTEM FOR COMMUNICATING CASHLESS VENDING TRANSACTION DATA AND VENDING MACHINE AUDIT DATA TO REMOTE LOCATIONS, filed January 18, 2002; which is a continuation in part application that claims priority of a U.S. non-provisional application, Serial No. 09/888797, inventor H. Brock Kolls, entitled A METHOD OF PROCESSING CASHLESS PAYMENT TRANSACTIONS WORLDWIDE, filed June 25, 2001; which is a continuation in part application that claims priority of a U.S. non-provisional application, Serial No. 09/884755, inventor H. Brock Kolls, entitled SYSTEM FOR PROVIDING REMOTE AUDIT, CASHLESS PAYMENT, AND INTERACTIVE TRANSACTION CAPABILITIES IN A VENDING MACHINE, filed June 19, 2001.

## TECHNICAL FIELD OF THE INVENTION

The present invention relates to a system and method that is scalable and configurable to include interfaces for vending equipment monitoring and control capabilities, interfaces for a card reader device and other identification devices as payment for items vended, an interactive interface and protocol for interconnecting the system to a computing platform, and support for a plurality of communication options that include wired, point-to-point wireless, and wireless networking including LAN/ WAN/ WCDMA/ CDMA/ CDPD/ 2G/ 2.5G/ 3G solutions. In an exemplary embodiment the system can be embodied in a semiconductor package or module package.

In addition, the present invention also relates to a system and method of effectuating a payment device for accepting card ID data, authorizing the validity of the card ID data, facilitating a vending transaction, settling the transaction to effect payment for the vended goods and services, gathering DEX and MDB audit data from the vending equipment, and data communicating with a plurality of remote locations where a remote location can be a global network based data processing resource.

## BACKGROUND OF THE INVENTION

Recent trends in the vending industry have been to offer higher priced items out of vending equipment at traditionally unattended vending locations. Higher priced item offers can result from the desire to vend larger portions of products such as the twenty-ounce soda bottle verse the twelve-ounce soda can. In other cases the higher priced items can be items that until recently my not have been consider for sale through vending equipment such phone cards, disposable cameras, and frozen food entrees to name a few.

The vending industry's desire to vend higher priced items has given rise to issues related to currency and inventory. For example, with the shift to vending twenty-ounce bottles many of the vending sales now involve more than one currency note, as an example two one-dollar bills to make a purchase. As a result the bill validator can fill to capacity with currency notes before all the items in the vending equipment have been sold. With a bill acceptor filled to capacity the vending equipment may not be able to transact another vending sale and place itself out-of-service. As a result vending operators can typically find themselves restocking vending equipment that still has product available for sale but because of the inability to take additional currency notes the vending equipment could not sell the inventory.

In addition, with many beverage type vending machines the shift from the twelve-ounce can to the twenty-ounce bottle can create coin mechanism issues. In moving to the larger size beverage the average price can move from typically slightly less than a dollar where little change was required when a dollar note was used for payment to slightly more than a dollar where the better part of a dollar in change can be required when two one-dollar notes are used for payment for a vend. Resultant from this price move not only do the bill validators fill to capacity faster and stop working sooner, but the coin mechanism can be required to supply a customer with more change on each vend depleting a coin mechanism coin supply faster. Once the coin change supply is depleted the vending machine may be rendered out-of-service.

In addition to the new burden on bill acceptors and coin mechanisms resultant from the sale of higher priced items other issues related to the vending of higher priced items can arise. One such issue can be that a customer may not have enough money on-hand to effectuate the vending purchase. In the case of phone cards, cameras, and frozen food vend prices may range from several dollars to forty dollars, fifty dollars, or more. In many cases the customer may have the desire to purchase the high priced item but simply

Such telecommunication costs can include the cost of running a telephone line to the vending equipment. In many cases the vending equipment may be in a location not





The present invention also relates to a system and method of effectuating a payment device for accepting card ID data, authorizing the validity of the card ID data, facilitating a vending transaction, settling the transaction to effect payment for the vended goods and services, gathering DEX and MDB audit data from the vending equipment, and data communicating with a plurality of remote locations, where a remote location can be a global network based data processing resource.

The present invention also relates to the system 500 being packaged in semiconductor or module creating a single chip or single module system 500 solution.

The single chip or single module system 500 solution can be referred to as semiconductor 500. The functionality of semiconductor 500 can include at least one of the following: cashless payment functionality, network connectivity functionality, or digital content presentation functionality.

5

## BRIEF DESCRIPTION OF FIGURES

The present invention is best understood from the following detailed description when read in connection with the accompanying drawings. Included in the drawings are the following Figures:

Figure 1A-1B there is shown a vending machine interface unit 100;

Figure 2A-2B there is shown a transceiver and modem base unit 200;

15

Figure 3A there is shown a front view of a card reader assembly;

Figure 3B there is shown a left side view of a card reader assembly;

20

Figure 3C there is shown a right side view of a printer assembly;

Figure 3D there is shown a front view of a printer assembly;

25

Figure 3E there is shown a right side view of a card reader assembly and a right side view of a printer assembly being aligned for assembly together;

Figure 3F there is shown a right side view of the assembled card reader and printer assembly;

Figure 3G there is shown a front view of a payment module;

Figure 3H there is shown a left side view of a payment module;

5

Figure 3I there is shown a front view of a payment module with receipt printer slot;

10

Figure 3J there is shown a left side view of a payment module with display and communications board included;

Figure 3K there is shown a right side view of the payment module assembly and a right side view of a printer assembly being aligned for assembly together;

15

Figure 4 there is shown a vending machine, vending machine interface unit, card reader and printer assembly, and transceiver and modem base unit;

Figure 5 there is shown an audit-credit-interactive system 500;

20

Figure 6A there is shown card reader and user interface system 600;

Figure 6B there is shown a card reader and user interface system 600 data communication routing switch.

25

Figure 7 there is shown a transceiver and modem base unit system 700 and a plurality of remote locations;

Figure 8 there is shown an audit-credit-interactive system 500 interfaced to a computing platform;

5 Figure 9A there is shown a vending machine MDB interface with a plurality of peripheral devices;

Figure 9B there is shown an audit-credit-interactive system 500 interfacing to a vending machine MDB bus and interfacing to a plurality of peripheral devices by way of a system 500 mimic MDB bus;

10

Figure 9C there is shown an audit-credit-interactive system 500 with card reader and audit functionality embodiment interfacing to a vending machine MDB bus and interfacing to a plurality of peripheral devices by way of a system 500 mimic MDB bus;

15

Figure 9D there is shown a MDB TRANSACTION STRING with system 500 and vending equipment interface;

20 Figure 10A-10B there is shown a system 500 semiconductor package, and a system 500 module package;

Figure 10C-10D there is shown an audit-credit-interactive system 500 embodied in a semiconductor package;

25 Figure 11 there is shown an MDB initialization tuning routine 1100;

Figure 12A-12B there is shown a vending machine interface unit (VIU) 100 with system 500 and transceiver and modem base unit system 700 wireless protocol data communication routine 1200;

5 Figure 13 there is shown a local transaction authorization routine 1300;

Figure 14 there is shown an international transaction authorization and settlement routine 1400;

10 Figure 15 there is shown a data communication transaction message parsing routine 1500;

Figure 16A-16B there is shown a determination of transaction completion routine 1600;

15 Figure 17 there is shown a data communication sweeping, processing, and data forwarding routine 1700;

Figure 18A-18B there is shown a mimic MDB interface port routine 1800;

20 Figure 19A-19B there is shown a local authorization database management routine 1900;

25 Figure 20 there is shown a transceiver and modem base unit system 700 wireless protocol data communication routine 2000; and

Figure 21 there is shown a MDB TRANSACTION STRING updating routine 2100.



## DETAILED DESCRIPTION OF THE INVENTION

Referring to Figure 1A and 1B there is shown a vending machine interface unit (VIU) 100. Figure 1A shows an antenna 124 mounted perpendicular to the enclosures main body. Figure 1B shows the antenna 124 mounted parallel to the enclosure main body. In an exemplary embodiment, antenna 124 is included in embodiments making use of the wireless VIU 100 connectivity and excluded in embodiments not requiring wireless VIU 100 connectivity.

10

The VIU 100 is a control system that interfaces to a plurality of different kinds of vending machines by way of a plurality of different interface ports. One such interface port can be the NATIONAL AUTOMATED MERCHANDISING (NAMA) vending industry MULTI-DROP-BUS (MDB) interface. Other MDB interfaces can include derivative MDB bus specifications, where a derivative MDB bus specification can be one that support less than all the NAMA standard, or augments the NAMA standard with additional protocol commands and or features. A second such interface port can be the EUROPEAN VENDING ASSOCIATION'S vending industry DATA EXCHANGE INTERFACE (DEX) interface. Additional interface ports include serial and pulse style bill validators, and coin mechanism interfaces.

20

Vending machine types suitable for interconnection to and operation with the VIU 100 include vending beverage and snack machines, value adding equipment, and dispensing equipment that operates in connection with or makes available a MDB bus interface, or DEX interface, or a bill acceptor interface, or a coin mechanism interface. Such vending machines include for example and not limitation those manufactured by or for COKE-A-COLA, PEPSI, MARS, VENDO, ROYAL, DIXIE NARCO, GPL, CRANE NATIONAL, AUTOMATED PRODUCTS, CAVALIER, MARCONI or other similar

25

vending machines. Such value adding equipment and dispensing equipment can include for example and not limitation those manufactured by or for ACT, XCP, SCHLUMBERGH, DAYNL, DEBITEK, GILBARCO, MARCONI, COPICO, PRE-PAID EXPRESS, or other similar value adding equipment and dispensing equipment.

5

For purposes of disclosure the term vending machine, value adding machine, and value dispensing machine can be referred to as a vending machine, vending equipment, and or vender. Other vending machine can include beverage style vending machines, snack style vending machines, specialty style vending machines, copiers, fax machines, personal computers (PC), data ports, office equipment, and or other types of vending, retail, office products, or business center types of equipment. Specialty style vending machines include for example and not limitation ice cream vending machines, amusement and arcade games, amusement ride game commonly found in store fronts and shopping malls, fresh produce machines, french fry vending machines, novelty product vending machines, consumer goods style vending machines, and or services type vending machine (such as name tag making, card making, polishing machines, and other service types of vending machines).

Audit-credit-interactive system 500 electronics are included within the VIU 100. For purposes of disclosure a system 500 can be referred to as a vending machine interface unit, VIU 100, or audit-credit-interactive system 500. Many of the electrical interfaces, ports, and connectors shown in Figure 1A and 1B are actually electrical connection to the audit-credit-interactive system 500 (system 500).

The vending machine interface unit (VIU) 100 includes an interactive interface port 102. The interactive interface port 102 provides an electrical connection to the interactive interface 532. In an exemplary embodiment the interactive interface port 102 enables other computing platforms to interface to and operational work with the vending

5

20

25

industry specification entitled MDB/ICP INTERFACE PROTOCOL version 1.0 and version 2.0. The electrical characteristics and operation of the DEX port 112 are detailed in the EUROPEAN VENDING ASSOCIATIONS EVA-DTS specification version 4.0, and 5.0. MDB interface 518, DEX interface 520, bill and coin interface 506, mimic MDB  
5 interface 516, and office products interface 534 can be referred to as peripheral device interfaces.

The MDB interfaces allow the VIU 100 by way of the MDB interface 518 and MDB port 108, to be original equipment manufactured (OEM) into or retrofitted into  
10 vending, valuing, and dispensing equipment that provide an MDB bus interface. Furthermore, the VIU 100 by way of the DEX interface 520 and the DEX port 112, can be original equipment manufactured (OEM) into or retrofitted into vending, valuing, and dispensing equipment that provide a DEX interface.

15 VIU 100 includes card reader interface ports 110, and 114. The card reader ports 110, and 114 provide electrical connection to the card reader interface 526. Card reader interface ports interface to industry standard bit strobe, and serial style track 1, 2, and 3 card readers. Such card readers include for example and not limitation those manufactured for or by XICO, NEURON, MAGTEK, as well as compatible card readers  
20 manufactured by other companies.

The VIU 100 also includes an RJ11 jack 116. The RJ11 jack provides electrical connections to the modem 522. In an exemplary embodiment the RJ11 jack 116 interconnects the VIU 100 to a telecommunication line, wherein data communication can  
25 occur between the VIU 100 and a plurality of remote hosts networks and locations.

VIU 100 also includes a general-purpose input-output interface 118. The general-purpose input-output interface provides electrical connections to the bill and coin

interface 506. In an exemplary embodiment the VIU 100 can be interconnected with vending, valuing, and dispensing equipment by way of the host equipment's bill acceptor or coin interface port. This allows the VIU 100 by way of the bill and coin interface 506 and interface 118 to be original equipment manufactured (OEM) into or retrofitted to  
5 vending, valuing, and dispensing equipment that utilize a serial or pulse style bill acceptor, or a coin mechanism interface. Serial and pulse style bill acceptors include for example and not limitation those manufactured for or by MARS, COINCO, CONLUX, ARDAK, or other similar bill acceptor and manufacturers of bill acceptors.

10           The VIU 100 includes a service button 120 and a ground terminal 122. The service button provides one of a plurality of electrical connection to the keypad and button inputs 510. The ground terminal 122 provides, as may be required, electrical connection to the VIU 100 enclosure.

15           Antenna 124 can pass through the VIU 100 enclosure or be mounted to the VIU 100 enclosure. The antenna 124 provides an antenna electrical connection to the transceiver 524, data modem 514, or optionally an antenna electrical connection to an external modem interconnected with auxiliary interface port 106.

20           Referring to Figure 2A and 2B there is shown a transceiver and modem base unit 200. Transceiver and modem base unit 200 includes transceiver unit 700 built in. The transceiver unit 200 with transceiver unit 700 data communicates wirelessly with the VIU 100 and by way of a modem data communicates with a remote location. In an exemplary embodiment the VIU 100 with system 500 and transceiver unit 200 with transceiver unit  
25 700 form a wireless data link, which has access to a modem for data communicating with a remote location. In this regard the reliance on having a telecommunication line in proximity to the VIU 100 or more generally in proximity to the vending equipment the VIU 100 is installed in is greatly reduced.

The transceiver system 700 located inside the transceiver unit 200 enclosure can obtain power for operation from an electrical connection by way of AC connection 208. In an exemplary embodiment the AC connection 208 can be plugged into a standard 115VAC wall outlet.



A push button switch 308 can be fastened to the faceplate 302 and electrically connected to the card reader interface board 312 by way of cable assembly 336. In



addition, card reader 310 can be electrically connected to the card reader interface board 312 by way of cable assembly 314.

Referring to Figures 3C and 3D there is shown a printer assembly. Figure 3C  
5 shows a right side view of the printer assembly. Figure 3D shows the front view of the printer assembly. In an exemplary embodiment the printer assembly can be slid onto the card reader assembly (see Figure 3F) and secured to the card reader assembly by way of the thumbscrew 334. A card reader assembly having been equipped with a printer  
10 system.

At the top of printer bracket 330 there is a cutout for receiving a paper holder rod 324. The paper holder rod 324 is typically inserted through a roll of paper, such as paper roll 322. Printer bracket 330 also has a plurality of mounting holes to secure printer the  
15 mechanism 328 to the printer bracket 330. Printer bracket 330 has a threaded thumbscrew 334 secured to the bracket 330.

The printer mechanism 328 has a paper advance knob 326. The paper advance knob 326 can be used to position the paper.

20

Referring to Figure 3E there is shown a right side view of a card reader assembly and a right side view of a printer assembly being aligned for assembly together. In an exemplary embodiment a user of the card reader assembly can choose to add the ability to print receipts, coupons, and other print information by sliding the printer assembly  
25 onto the card reader assembly and making the appropriate electrical connections. Furthermore, the printer assembly can be securely fastened to the card reader assembly by way of thumbscrew 334.

Referring to Figures 3G-3H there is shown a payment module assembly. In an exemplary embodiment the payment module assembly can be installed in a vending machine's dollar bill validator slot. A user having access to the front of the payment module assembly can insert cards, view transaction status information via the three LED display (340, 342, 344), and use a push button 308 to provide system input. A faceplate 302 is shown fastened to a support bracket 318. The faceplate 302 is sized to fit the industry standard bill validator opening, which can be found on most brands and models of vending equipment. The faceplate 302 has a plurality of holes to allow fastening of the

5

15

20

25

25

To allow functionality expansion of the base payment module which could include printing options, communication options, electronic locks, and other vending machine interface options (such as DEX) a daughter board could be added to the base

payment module. More specifically, additional payment module functionality through the use of one or more daughter cards could include: support for optional DEX interface, support for optional 16 character by 2 line LCD display or other types and styles of LCD or vacuum florescent displays, support for optional printer circuit card; and support for  
5 optional communication card (communication card could support point-to-point, modem, and external modem interfaces).

Referring to Figures 3I and 3J there is shown how a payment module with communications daughter board, and optional LCD display could be incorporated into the  
10 base level payment module. Referring to Figures 3I and 3J there is shown a payment module assembly. In an exemplary embodiment the payment module assembly can be installed in a vending machine's dollar bill validator slot. A user having access to the front of the payment module assembly can insert cards, view transaction status information via the LCD display or the three LED display (340, 342, 344), and use a push  
15 button 308 to provide system input.

A faceplate 302 is shown fastened to a support bracket 318. The faceplate 302 is sized to fit the industry standard bill validator opening, which can be found on most brands and models of vending equipment. The faceplate 302 has a plurality of holes to  
20 allow fastening of the card reader assembly into the vending equipment.

Faceplate 302 also has a paper exit slot 304 to allow a receipt printer (not shown) to dispense a printed receipt to a user of the system. Faceplate 302 also has a display slot 306 which allows the LCD display mounted on the threaded studs 346 to be viewable  
25 from its mounting location behind the front surface of faceplate 302. An LED display assembly comprising LED 340, 342, and 344 are viewable to a user. The LED display communicates status information related to the state of the vending machine or state of the card reader, as well as the status of a current transaction. The function of the LED

display and color of the LED can be selected or defined in accordance with EVA cashless payment guidelines for LED displays.

5 An interface connection 348 on the payment module board 350 provides an electrical interface to external data processing equipment and or other computing platforms. Such external data processing equipment and or computing equipment can include communication devices, display devices, telemetry devices, VIU's, interactive media devices, PC based platforms, and other types of computing platforms. Interface 348 can be an external peripheral interface 536, a network interface 542, an interactive 10 interface 532, or other type of interface.

In addition, faceplate 302 can be fastened to a bracket 318. Bracket 318 has a plurality of bracket fasteners for fastening a card reader 310 to the card reader assembly. The bracket 318 also has a threaded insert 316 located in the rear of the bracket 318. 15 Threaded insert 316 can receive thumbscrew 332 in order to facilitate the fastening of the printer assembly bracket (not shown) to the card reader assembly. An expansion board 354 can be mounted to the bracket 318 and electrically interconnected to the base level payment module board 350 by way of cable assembly 356 or as a PCB mountable component.

20

A push button switch 308 can be fastened to the faceplate 302 and electrically connected to the payment module board 350 by way of cable assembly or as a PCB mountable component. LED display 340, 342, and 344 can be electrically connected to the payment module board 350 by way of cable assembly or as a PCB mountable 25 component. In addition, card reader 310 can be electrically connected to the payment module board 350 by way of cable assembly or as a PCB mountable component.



Referring to figure 3K there is shown a payment module and printer assembly being aligned for assembly together. In an exemplary embodiment support for a printer option can be accomplished with the addition of a daughter card or interconnection of a printer mechanism with the base level card reader board 350, and the addition of the  
5 printer assembly. Shown in Figure 3K is a printer assembly plate 330, interconnected with a paper roll holder 324, a printer mechanism 328, and a thumbscrew fastener 332. Also shown is a paper roll 322 being held by the paper roll holder 324 and a paper feed adjuster 326, which is part of printer mechanism 328.

10 In an exemplary embodiment a user of the payment module assembly can choose to add the ability to print receipts, coupons, and other print information by sliding the printer assembly onto the card reader assembly and making the appropriate electrical connections to the daughter board 226 or card reader card 350. Furthermore, the printer  
15 assembly can be securely fastened to the card reader assembly by way of thumbscrew 332.

Referring to Figure 4 there is shown a vending machine 402, vending machine interface unit 100, card reader with optional printer assembly 406, and transceiver and modem base unit 200.

20

In an exemplary embodiment a VIU 100 can be located inside the vending equipment, such as vending equipment 402. In addition the card reader assembly with optional printer assembly can be mounted inside the vending equipment in such a way that a user has access to the card reader assembly. During operation a communication line  
25 can be interconnected directly with the VIU 100. Alternatively the VIU 100 can wireless data communicate with a transceiver base unit 200. There is shown in Figure 4 a transceiver unit 200 plugged into an electrical outlet on wall 202. Also shown is a telecommunication line 408 interconnect with transceiver unit 200.



The audit-credit-interactive system 500 includes numerous mutually exclusive interfaces and control means. In a plurality of customer specifications and where

25 Interconnected with microcontroller 502 can be bill acceptor and coin mechanism interface 506. The bill acceptor and coin mechanism interface 506 emulate industry standard bill acceptor and coin mechanism interfaces. In this regard the audit-credit-interactive system 500 can be interconnected to vending equipment by way of the

interface 506. The audit-credit-interactive system 500 mimicking industry standard bill acceptor and coin mechanism electrical control system and signal timing can then operate the vending equipment. Industry standard bill acceptors include serial and pulse style. Serial style bill acceptors utilize INTERRUPT, SEND, ACCEPT ENABLE, and DATA control signal lines. Pulse style bill acceptor and coin mechanism send electrical pulses to an attached control system to indicated the receipt of coin and currency. Serial and pulse style bill acceptors and coin mechanisms can include for example and not limitation MARS, COINCO, CONLUX, or other similar bill acceptors and or coin mechanisms.

Interconnected with microcontroller 502 can be a display interface 508. A display interface 508 can be a liquid crystal display (LCD), a vacuum florescent display, an RS232 connection, and or an electrical interface for driving a display. In an exemplary embodiment display interface 508 can be, for example and not limitation, an RS232 serial connection. Such a serial connection can be utilized to data communicate display data as well as other types of data to a card reader interface board 312.

Interconnected with microcontroller 502 can be a plurality of keypad and button inputs 510.

Interconnected with microcontroller 502 can be memory 512. Memory 512 can be a plurality of different types of memory. In an exemplary embodiment memory 512 can comprise non-volatile random access memory (NOVRAM), random access memory (RAM), flash memory, and serial flash memory. In addition, the RAM/NOVRAM can include a timekeeper function for maintaining date and time. RAM/NOVRAM can be a DALLAS SEMICONDUCTOR DS1644, DS1646, or DS1647, or other similar or suitable RAM. Flash memory can be an ATMEL or STS brand AT29E010 or other similar style, different size, other brand, or suitable substitute. The serial flash memory

can be an ATMEL brand AT45D081, a MICROCHIP 93LC66, or other similar style, different size, other brand, or suitable substitute.

5 In an exemplary embodiment the timekeeper feature can be effectuated to time and date stamp the transactions as they occur. From the vending equipment's MDB interface CASH VEND transactions, if supported by the VMC, as well as cashless vend transactions can be monitored and recorded. Adding a time and date time stamp to the each transaction as they occur can result in a detailed inventory utilization record showing the date and time the products were vended.

10

Interconnected with microcontroller 502 can be an office product interface 534. An office product interface 534 can include, for example and not limitation, an optoisolator for counting pulses generated by a fax machines, copy machine, and other office product equipment. In addition, office product interface 534 can include, for  
15 example and not limitation, a DTMF decoder for decoding telephone touch-tones and subsequently billing for the use of a telephone line. DTMF decoding can be used in connection with a fax machine to bill for usage based in part on local, long distance, and international dialed locations.

20

Interconnected with microcontroller 502 can be an external peripheral interface 536. The external peripheral interface 536 includes a plurality of configurable input and output line for interfacing to external peripheral devices. External peripheral interface 536 can support serial peripheral interfaces (SPI), serial interfaces such as RS232, RS485, I<sup>2</sup>C, and other types of peripheral interfaces and communication protocols and  
25 standards.

Interconnected with microcontroller 502 can be a network interface 542. A network interface 542 can be an Ethernet, token ring, FIREWIRE, or other similar or suitable type of network interface.

5           A data modem 514 can be interconnected with microcontroller 502. A data  
modem 514 can effectuate wired and wireless data communications with a plurality of  
remote locations. Wireless data modems include, for example and not limitation,  
MOTOROLA, ERICKSON, and NOKIA brands of data modems, as well as SPRINT  
PCS, CDMA, CDPD, 2G type wireless device, 3G type wireless device, research in  
10 motion (RIM) type wireless device, or other similar or suitable brands or types of  
wireless data modem.

A multi-drop-bus (MDB) interface 518 is interconnected with microcontroller 502. In an exemplary embodiment an MDB interface 518 electrically interconnects with the vending equipment's MDB bus. An MDB interface 518 can be implemented with a universal asynchronous receiver transmitter (UART). In addition an MDB interface 518 can include a buffer circuit to handle the physical interface requirement to connect the system 500 to the MDB bus. A suitable buffer circuit can be an opto-isolated circuit or other suitable buffer circuit interface.

20 In an exemplary embodiment the UART can be configured to operate with eight data bit and one address bit in addition to start and stops bits. In this regard in accordance with NAMA MDB specifications the MDB interface 518 can operate in and support the multi mode - multi processor addressing requirements. In operation the UART under  
25 control of the microcontroller 502 detects a valid address byte being data communicated from the VMC. The valid address byte indicates to the system 500 that the data following from the VMC should be captured and stored, parsed, responded and or in general the system 500 should data communicate with the VMC as appropriate.

25 In an exemplary embodiment a VMC can include peripheral device interface support for MDB interface 518, DEX interface 520, bill and coin interface 506, and office products interface 534.



In an exemplary embodiment, to handle MDB communications an MDB message routine can be implemented in firmware under the control of the microcontroller 502. In this regard a series of interrupt routines can be utilized to first detect the message and then trigger a routine to parse and respond to the received message. A UART receive routine can be effectuated to capture and store data being communicated from the VMC.



20 A second advantage of the dual mode of operation of the mimic MDB interface  
516 is that features supported by a peripheral resultant from the implementation of a  
derivative MDB specification can be utilized by data communication first to the system  
500 by way of the mimic MDB interface 516. If the MDB protocol command is a  
command supported by the vending equipment's VMC controller the system 500 can  
25 then relay the message received from the peripheral device to the VMC control system by  
way of the system 500's MDB interface 518. In this regard the system 500 essentially  
acts as a MDB interface gateway sending and receiving non-VMC support portions of a  
peripheral's implement MDB specifications. In addition, the MDB gateway implemented

by the system 500 can allow the peripheral device data communication access to the VMC controller for portions of the peripheral implemented MDB specification supported by the vending equipment's VMC controller.

5           A data exchange (DEX) interface 520 is interconnected with microcontroller 502. The DEX interface 520 is a serial connection interface for interfacing the system 500 to the VMC control system. In an exemplary embodiment the DEX interface conforms to the EVA-DTS version 4.0 and version 5.0 specifications. In this regard the system 500 can 'DEX' vending equipment and obtain marketing, sales, and operational data as well  
10 as other types of data related to the vending equipment operation and performance. In addition, the DEX interface 520 can be utilized to program the VMC control system. VMC programming can include setting prices and parameters, setting operational data, clearing error codes or messages, and programming the VMC firmware.

15           In an exemplary embodiment DEX interface 520 can be configure as a transistor-to-transistor level (TTL) or RS232 style serial connection. Furthermore, the transmit line from the system 500 can be independently signal level configurable such that during non-data communication idle time the signal level of the system 500 DEX transmit line can be set to a high impedance state or a low signal level state as oppose to the tradition idle  
20 high signal level state. The advantage can be the most VMC equipment is configure to detect the insertion of a plug into the DEX port. The insertion of the plug places a high signal level from the DEX port transmit line onto the VMC receive line. This can invoke the VMC to begin a DEX session with the device plugged into the DEX port. The ability to selectively configure the non-communication idle state signal level of the system 500  
25 DEX transmit line to a high impedance state or low signal level state allows the system 500 DEX connection to be made with the VMC DEX port without the DEX port detecting the connection has been established. In this regard the system 500 only needs to change the state of the system 500 DEX transmit line to trigger a DEX session with the

25

25           An interactive interface 532 can be interconnected to microcontroller 502. The interactive interface 532 can be utilized in combination with the interactive interface communication protocol shown in the table below to interconnect the system 500 to a computing platform. The card reader assembly having a card reader interface board 312,

which is implementing a card reader user interface system 600, is a computing platform. In addition, PC based devices, handheld devices, and other microprocessor-based devices are also computing platforms.

5 In an exemplary embodiment for example and not limitation a computing device can be interfaced to a system 500 by way of the interactive interface 532. In this regard the two interconnected devices can data communicate by way of an interactive device interface protocol. This protocol can be implement in and exemplary embodiment as disclosed in the following table as an example and not limitation where a system 500 can  
10 be referred to as a MDB controller or G4, and host network center 808 can be referred to as USALIVE:

---

15 **MDB Controller/G4 E-port/E-port (System 500) Interactive Interface  
Communication Protocol Interface And Specification**

**MDB Controller/G4:**

20 The MDB controller is the microcontroller-based system, which can interfaces to the vending machines MDB interface and to a computing platform. Such computing platforms include E-port. Certain versions of the E-port may incorporate the MDB controller and computing platform into a single board solution. In such a case serial communications between the computing platform and the MDB microcontroller occur over the devices serial peripheral interface (SPI), serial interface or other similar or  
25 suitable communication interface.

The G4 (system 500) version of E-port can utilize a single microcontroller to serve as an MDB controller or a semiconductor module as well as a cashless payment

system platform. The G4 device incorporates an RS232 serial interface by which other computing platforms can interface to and control the functionality of the G4 and associated vending equipment. The G4 version can operate in two modes of operation. In a first mode of operation the G4 provides all the MDB interface control, audit/cashless payment support, and network connectivity. In this mode a computing platform can interact with the G4 in a hybrid role to monitor a string of user text prompts (see DISPLAY PROTOCOL) as well as execute NON-MDB-CONTROL types of commands (see disclosure below).

10 In a second mode of operation the G4 can be configured and serve as an MDB controller (system 500) only. In this mode both the MDB-CONTROL and NON-MDB-CONTROL commands can be executed. While in this mode of operation of the computing platform operates as a master device controlling the operation and process flow of the system. While in this mode the G4 serves as a slave device interfacing to the vending machine and managing the control of the MDB interface. COMMUNICATION INTERFACE details the electrical interconnections required to allow the G4 to data communicate with a computing platform.

#### Communication Parameters:

20

The MDB controller/G4 communicates to a computing platform by way of serial communications. In this regard a set of commands issued from the computing platform implement a level of control via the MDB controller/G4 and the MDB/ICP protocol to transact a cashless transaction, obtain DEX data information, and other vending machine related data. The communication protocol implemented between the system 500 (also referred to as the G4, G4 eport, or MDB controller) and the computing platform can be referred to as the Interactive Interface Communication Protocol.

25

Serial communications between the computing platform and the MDB controller/G4 are set at 9600 baud, 8 data bits, No Parity, and 2 Stop bits. Required serial port communications lines include transmit (Txd), Receive (Rxd) and Ground (Gnd).

5

## MDB CONTROLLER/G4 COMMAND PROTOCOL

### Protocol Exchange:

- 10 The “master” computing platform can initiate any ‘@’, ‘#’, ‘AAA’, or ‘BBB’ command listed in the command disclosure below. Commands are case sensitive. In response to a complete command the MDB controller/G4 will process and return the result string. The result string shall start with a start character (STX) hex \$02, and conclude with an ETX character hex \$03. A LRC check byte will immediately follow the
- 15 ETX character.

STX	hex \$02
ETX	hex \$03
<esc>	ESCAPE character hex \$1B
LRC	all bytes XORed excluding the STX character and including the ETX character.

### Protocol Exchange Characters

- 20 It is recommended that the ‘@’ commands be executed by inserting a leading space prior to the ‘@’. For example sending ‘ @<esc>H’ instead of ‘@<esc>H’ the differencing being a leading space. The leading space will decrease command communication errors by allowing the MDB Controller/G4 to sync on the leading space.



**MDB-CONTROL-COMMANDS****Command Description:**

5    @<esc>T – REQUEST FOR CARD READER DATA.

If no card reader data is available then the result string will return:

STX+[CARD-NOCARDDATA]+ETX+LRC

10

Else the result string will return:

STX+[CARD-][UPTO 37 BYTES MAX OF CARD DATA]+ETX+LRC

15    Example of valid card data:

STX+[CARD-41324132413243132=9182738937463764372]+ETX+LRC

20    @<esc>Z – CLEAR CREDIT CARD DATA. The credit card buffer will be cleared and  
The result NOCARDDATA will be inserted in the buffer and returned in response to the  
@<esc>T command. The result string will return.

STX+[OK-Z]+ETX+LRC

25    @<esc>V – REQUEST FOR MDB TRANSACTION STRING DATA. Referring to  
Figure 9D there is shown a MDB TRANSACTION STRING with system 500 and  
vending equipment interface.

25

Microcontroller having data communication access to memory 512A constructs and manages the MDB transaction string in memory. As data is exchanged between the vending machine 402 and the system 500, microcontroller 502 updates the MDB transaction string as required.

Utilization of the MDB transaction string occurs when application code running in memory 512B, and or computing platform 802 interconnected with system 500 by way of the interactive interface 532 read the MDB to make certain determinations. For example and not limitation an application running in memory 512B can read the MDB string VEND STATE FLAG field and determine the current state is MDB VEND STATE 'D' for disable. This can cause the application code to display a message by way of display interface 508 that the vending machine is DISABLED and can not transact a vend. Upon the vending machine 402 data communicating with the system 500 by way of the various vending equipment interfaces the vending machine 402 may communicated the MDB reader enable 14 01 (as defined in the NAMA MDB specification) command. The microcontroller 502 would then update the MDB TRANSACTION STRING with the 'E' for enable. The application code running in memory 512B upon reading the VEND STATE FIELD would now determine the vending machine is MDB VEND STATE 'E' for enabled and could display a message by way of display interface 508 that the vending machine is ENABLED and ready for vending.

In another exemplary embodiment a computing platform 802 can monitor the MDB TRANSACTION STRING to determine the current state and condition of the vending machine 402. In addition, the computing platform 802 can data communicate by way of the interactive interface 532 commands to be executed by microcontroller 502. Such commands are referred to as the '@', '#', AAA, and BBB commands disclosed herein.

As an example and not limitation, computing platform 802 can monitor and control vending machine 402 by way of interactive interface 532 by utilizing the @<esc>I, @<esc>V, @<esc>T, and or the @<esc>H. commands to monitor the presence of card data and the current MDB TRANSACTION STRING setting. One advantage of

invoking the @<esc>I interrupt command is the once invoked any time the MDB TRANSACTION STRING of card reader buffer change the MDB TRANSACTION and or card reader buffer will be data communicated from the system 500 to the computing platform 802. In this configuration the computing platform 802 does not need to poll to  
5 determine whether card data or MDB TRANSACTION STRING status has changed – the data will be sent to the computing platform 802 when it does.

The computing platform 802 can determine from the MDB TRANSACTION STRING that the vending machine is MDB VEND STATE 'E' enabled and ready for a  
10 vend cycle. The computing platform can then data communicate to the system 500 one of the @<esc>S, @<esc>B, or one of the @<esc>A (session commands) commands. These command instruct the system 500 to take certain actions that can include having the system 500 by way of the vending equipment interfaces data communicated to vending machine 402 the MDB BEGIN SESSION 13 04 (as defined in the NAMA MDB  
15 specification) command. The vending machine will responds by starting a session. The microcontroller 502 will update the MDB TRANSACTION STRING to reflect that the vending machine 402 is in MDB VEND STATE FLAG to 'S' session.

When a user pressing a button on the vending machine 402, the vending machine  
20 402 data communicates an MDB VEND REQUEST 13 05 (as defined in the NAMA MDB specification) with SALE PRICE and COLUMN detail information attached. System 500 upon receipt of the can update the MDB TRANSACTION STRING to reflect the SALE PRICE, COLUMN, and VEND FLAG 'R' vend request. The computing platform can either read the MDB TRANSACTION STRING (non interrupt  
25 mode) or the MDB TRANSACTION STRING will be data communicated upon change to the computing platform 802 (interrupt mode). If the VEND ASSIST ON mode the computing platform 802 will have to send the #<esc>Q command to deny the vend request or the #<esc>R with sales detail to approve the vend request. The system 500 will

respond to the computing platform 802 #<esc>Q and #<esc>R command by updating the MDB TRANSACTION string accordingly and sending the VEND DENIED 06 (as defined in the NAMA MDB specification) command to the vending machine 402, or the VEND APPROVED 05 (as defined in the NAMA MDB specification) to the vending  
5 machine 402 respectively.

In the VEND ASSIST OFF mode the VEND APPROVED 05 (as defined in the NAMA MDB specification) may be automatically generated and sent to the vending machine 402 by the system 500 in response to the VEND REQUEST message from the  
10 vending machine 402.

In the event of the VEND APPROVED being sent to the vending machine 402 in response to a VEND REQUEST the MDB TRANSACTION STRING will be update to MDB VEND FLAG 'P' for vend pending while the vending machine is dispensing the  
15 goods and or services and then either an MDB VEND FLAG 'V' for vended or 'F' for vend failed. The computing platform 802 can read the MDB TRANSACTION STRING and take the appropriate actions including sending the @<esc>C command to clear the MDB TRANSACTION STRING to prepare for the next vend cycle.

20 Referring to Figure 21 there is shown a MDB TRANSACTION STRING updating routine 2100. Processing begins in block 2102.

In block 2102 the MDB TRANSACTION string is updated is system 500 memory. As previously described the updating occurs based in part on data  
25 communications between the system 500 and the vending machine 402, wherein the vending machine 402 is interconnected to the system 500 by way of the vending machine 402 VMC and the vending equipment interfaces 506, 516, 518, and or 520. Processing then moves to decision block 2104.

In decision block 2104 a determination is made at the system 500 as to whether computing platform 802 or application code running on microcontroller 502 is requesting the MDB TRANSACTION STRING be data communicated to the requesting destination.

5 If the resultant is in the affirmative that is the computing platform 802 or application code running on microcontroller 502 is requesting the MDB TRANSACTION STRING processing moves to block 2106. If the resultant is in the negative that is the computing platform 802 or application code running on microcontroller 502 is not requesting the MDB TRANSACTION STRING then processing moves to block 2108.

10

In block 2106 the command received from the computing platform 802 or the application code is received and the MDB TRANSACTION STRING is data communicated to the requesting destination. Processing moves back to block 2102.

15

In block 2110 the system 500 parses the received command. The various command can include the '@', '#', 'AAA', and 'BBB' detailed herein. Processing then moves to decision block 2112.

20

In decision block 2112 a determination is made as to whether the command the system 500 received required an MDB command be data communicated to the vending machine 402. If the resultant is in the affirmative that is the command received requires an MDB command be data communicated to the vending machine then processing moves block 2114. If the resultant is in the negative that is the command received does not require an MDB command be data communicated to the vending machine then

25

processing moves back to block 2102.



Field #1 is the MAX VEND PRICE as reported by the vending machine controller (VMC) during the MDB initialization process. The MAX VEND PRICE is the value of

Field #2 is the SALE PRICE. The SALE PRICE is determined in the MDB protocol for the VEND – Request Command (See. NAMA Multi Drop Bus (MDB)/Internal Protocol Version 1.0, and 2.0 specifications). The SALE PRICE is the vend sale price of the vend item selected as reported by the vending equipment during an MDB vend request transaction.

15

The 'F' field is the MDB transaction condition flag. Valid flag states include:

State	Description
C	Clear
\$	Currency vend has occurred
P	Vend Pending
A	Vend Approved
D	Vend Declined
V	Cashless vend has occurred
F	Vend fail

## Valid Flag States

STX+[V0001500001000002P]+ETX+LRC -> Vend state, MAX Vend price \$1.50, sale price \$1.00, vend from column 2, vend pending.

STX+[E0001500001000002V]+ETX+LRC -> Enable state, MAX Vend price \$1.50, sale price \$1.00, vend from column 2, vend complete.

- 5 STX+[E0001500001250003\$]+ETX+LRC -> Enable state, MAX Vend price \$1.50, sale price \$1.25, vend from column 3, cash vend.

STX+[S00015000000000003R]+ETX+LRC -> In session state, MAX Vend price \$1.50, vend request from column 3.

10

IMPORTANT NOTE: The MDB microcontroller/G4 MDB interface will continuously manage the changes to the MDB transaction string. For example as the MDB state changes, the MDB state field will automatically be updated. There are however two scenarios that require the execution of the @<esc>C – CLEAR MDB TRANSACTION

- 15 STRING DATA command. These two scenarios include when a currency vend has occurred the 'F' field has been set to '\$', and when a cashless vend has occurred and the 'F' field has been set to 'V'. In both these cases the @<esc>C command will have to be executed to clear the MDB TRANSACTION STRING before a new cash transaction can be tracked or a new cashless vending session can be started.

20

If the G4 is used in the VEND ACTIVE 'ON' mode the above does not apply in that the G4 will clear the MDB TRANSACTION string as appropriate. If the G4 is used in the G4 VEND ACTIVE 'OFF' the above will apply.

- 25 @<esc>C – CLEAR MDB TRANSACTION STRING DATA. The MDB controller/G4 will clear the SALE PRICE field, COLUMN information field, and the transaction condition flag is set to 'C'. The result string will return:

STX+[OK-C]+ETX+LRC

@<esc>H – HYBRID COMMAND FOR SEND CARD DATA AND MDB STRING.

- 5 The MDB controller/G4 will send both the card reader data (see @<esc>T above) followed by the MDB string (see @<esc>V). The result string will return:

STX+[ @<esc>T response]+ETX\_LRC+STX+[ @<esc>V RESPONSE]+ETX+LRC

- 10 @<esc>S – BEGIN A SESSION COMMAND. The MDB controller/G4 will begin an MDB session (see NAMA MDB specification V1.0, V2.0 for BEGIN SESSION command). The result string will return:

STX+[OK-S]+ETX+LRC

15

The G4 must have the MDB state set to 'E' for ENABLED in order to start a session. If a session cannot be started the result string will return:

STX+[UNABLE-S]+ETX+LRC

20

@<esc>X – END A SESSION COMMAND. The MDB controller/G4 will END an

MDB session (see NAMA MDB specification V1.0, V2.0 for SESSION CANCEL command). The result string will return:

25

STX+[OK-X]+ETX+LRC

@<esc>F – SET MDB CONTROLLER STATE TO INACTIVE. The MDB

controller/G4 will set the MDB state to Inactive. The result sting will return:

STX+[OK-F]+ETX+LRC

- 5 @<esc>**D** – SET MDB CONTROLLER STATE TO DISABLE. The MDB controller/G4 will set the MDB state to Disable. The result sting will return:

STX+[OK-D]+ETX+LRC

- 10 @<esc>**E** – SET MDB CONTROLLER STATE TO ENABLE. The MDB controller/G4 will set the MDB state to Enable. The result sting will return:

STX+[OK-E]+ETX+LRC

- 15 @<esc>**K** – PERFORM A HARDWARE MDB CONTROLLER RESET. The MDB controller/G4 will return the result string and then go through a hardware reset. The result sting will return:

STX+[OK-K]+ETX+LRC

20

@<esc>**I** – TOGGLE MDB INTERRUPT MODE. The MDB controller/G4 will return the result string below toggling between 'ON' and 'OFF' of the interrupt mode.

STX+[ON-I]+ETX+LRC -> When toggling into the interrupt mode

25

STX+[OFF-I]+ETX+LRC -> When toggling out of the interrupt mode

@<esc>**I** – turns mode 'ON'

STX+[ON-I]+ETX+LRC -> When toggling into the interrupt mode

@<esc>i – turns mode 'OFF'

STX+[OFF-I]+ETX+LRC -> When toggling out of the interrupt mode

5

While in the interrupt mode the MDB controller/G4 will send the result string for the @<esc>T and the @<esc>V commands shown above each time the respective data fields change.

10 For example, while in the interrupt mode the MDB controller/G4 will send the @<esc>T result string on the successful read of a magnetic card.

In addition, while in the interrupt mode the MDB controller/G4 will send the @<esc>V result string each time any field in the MDB transaction string changes.

15

Default: The default condition on microcontroller reset is 'OFF'.

@<esc>1 – TOGGLE MDB CODE CAPTURE MODE. The MDB controller/G4 will return the result string below toggling between 'ON' and 'OFF' of the MDB CODE

20 CAPTURE mode. This command is for diagnostic purposes only and should not be used during normal G4 operation. The intended purpose for this command is to diagnosis MDB related transaction issues during development and or testing.

STX+[ON-I]+ETX+LRC -> When toggling into the MDB code capture mode.

25

STX+[OFF-I]+ETX+LRC -> When toggling out of the MDB code capture mode

When the MDB code capture mode is switched to the 'ON' mode the following sequence of events begins:





'ON' capturing MDB transaction codes between the vending machine and the G4 while the vending machine and or G4 go through a power up or reset procedure.

If the G4 is in a vending transaction the TOGGLE MDB CODE CAPTURE MODE command cannot be executed to turn 'ON' the MDB capture feature. If the TOGGLE MDB CODE CAPTURE MODE command is executed during a vending session the MDB capture mode will be turned 'OFF' and result string will return:

STX+[OFF-1]+ETX+LRC

Default: The default condition on microcontroller reset is 'OFF'.

**@<esc>2 – MDB CAPTURE MODE BUFFER DUMP.** The MDB controller/G4 will return the result string below dumping the MDB codes passed between the G4 and the vending machine controller (VMC). The output will be formatted to indicate which codes were transmitted by the VMC and which codes were transmitted by the G4. This command is for diagnostic purposes only and should not be used during normal G4 operation. The intended purpose for this command is to diagnosis MDB related transaction issues during development and or testing.

STX+[MDB]+

-> Header

[VMC-]+VMC transmitted data

-> Data transmitted by the VMC

[G4 -]+G4 transmitted data

-> Data transmitted by the G4

• • • •

ETX+LRC

### **Command Description:**

USA-010-03

- 56 -

PATENT

@<esc>\$ – SIMULATE CASH VEND TRANSACTION. The G4 will simulate a CASH VEND transaction (see MDB spec V1.0 for CASH VEND command). The result sting will return:

5

STX+[OK-\$]+ETX+LRC

To simulate the CASH VEND the MDB transaction string will be set to the following:

10

STX+[E0005000001250001\$]+ETX+LRC

If the G4 is in a vending transaction a SIMULATE CASH VEND transaction cannot be executed. If a SIMULATE CASH VEND transaction cannot be executed the result string will return:

15

STX+[UNABLE-\$]+ETX+LRC

@<esc># – SIMULATE CASHLESS VEND TRANSACTION. The G4 will simulate a CURRENCY VEND transaction (see NAMA MDB spec V1.0 and V2.0 for VEND REQUEST and VEND APPROVED commands). The result sting will return:

20

STX+[OK-#]+ETX+LRC

To simulate the CASHLESS VEND the MDB transaction string will be set to the following:

25

STX+[E0005000001250001V]+ETX+LRC

If the G4 is in not in a vending transaction a SIMULATE CASHLESS VEND transaction cannot be executed. If a SIMULATE CASHLESS VEND transaction cannot be executed the result string will return:

5 STX+[UNABLE-#]+ETX+LRC

@<esc>M – TOGGLE MODEM COMMUNICATION ACCESS. The G4 will switch the serial communication ports being utilized by the computing platform to the MDB controller/G4 communication port. In this regard the computing platform can utilize the communication port (modem and or wireless) of the MDB controller/G4. The result sting will return:

15 STX+[ON-M]+ETX+LRC -> When toggling into the communication mode – allowing the computing platform to use the MDB controller/G4 communication port. A communication hardware reset will also be invoked in the G4 to prepare the communication device to receive data.

20 STX+[OFF-M]+ETX+LRC -> When toggling out of the communication mode

25 IMPORTANT NOTE: The communication parameters between the G4 are outlined above as 9600, no parity, 8 data bits, and 2 stop bits. Upon executing the @<esc>M command 'ON' the G4 switches direct access to the communication device. If for example, a 2400 baud modem is being used the device issuing the @<esc>M command will have to first change its baud rate to 2400 before the G4 modem can respond to the requests. Furthermore, upon the conclusion of a communication session before the @<esc>M command can be issued and interrupted by the G4 to switch the communications 'OFF' the baud rate of the device issuing the @<esc>M command should change its baud rate back to 9600 N, 8, 2.

RECORD TYPE	BYTES	DESCRIPTION
RECORD NUMBER	4 bytes	Current transaction record number
SEPARATOR	1 byte	Field separator ‘-‘
CARD DATA/ID DATA	37 bytes	Card data or Dial-A-Vend data
MERCHANT ID REFERENCE	1 byte	Merchant ID Prefix (G4 specific typically set to ‘1’)
SALE AMOUNT	5 bytes	Transaction sale amount
APPROVAL CODE	8 bytes	Transaction approval code (typically starts with AP)

USA-010-03

- 59 -

PATENT

CAPTURE ID FLAG	1 byte	Transaction Capture Flag/ Transaction ID 0=DO NOT CAPTURE TRANSACTION 1=CREDIT CARD TRANSACTION 2=SETTLEMENT DATA 3=ERROR RECORD 4=SETTLED CREDIT CARD TRANSACTION 5=PRIVATE SYSTEM TRANSACTIONS 6=NOT USED 7=EMAIL TRANSACTION
START TIME	8 bytes	Transaction Start Date and Time (MMDDHHMM)
COUNT 1	4 bytes	Event counter #1 i.e. copy or print count (XXXX)
STOP TIME	8 bytes	Transaction Stop Date and Time (MMDDHHMM)
COUNT 2	4 bytes	Event counter #2 i.e. copy or print count (XXXX)
INVENTORY TOTAL	2 bytes	Total vended inventory count
ITEM COLUMN 1	2 bytes	Vended item #1 column data
ITEM COLUMN 2	2 bytes	Vended item #2 column data
ITEM COLUMN 3	2 bytes	Vended item #3 column data
ITEM COLUMN 4	2 bytes	Vended item #4 column data
ITEM COLUMN 5	2 bytes	Vended item #5 column data
ITEM COLUMN 6	2 bytes	Vended item #6 column data
ITEM COLUMN 7	2 bytes	Vended item #7 column data
ITEM COLUMN 8	2 bytes	Vended item #8 column data

STX+[UNABLE-W]+ETX+LRC



**@<esc>R – TOGGLE VERBOSE TEXT PROMPTS ON/OFF.** The G4 will switch between providing a stream of text prompts (see DISPLAY PROTOCOL) when the VERBOSE mode is turned ‘ON’ and disabling the transmission of the text prompts when the VERBOSE mode is ‘OFF’. The result sting will return:

5

STX+[ON-R]+ETX+LRC -> When toggling into the VERBOSE mode.

STX+[OFF-R]+ETX+LRC -> When toggling out of the VERBOSE mode

10    @<esc>R – turns mode ‘ON’

STX+[ON-R]+ETX+LRC -> When toggling into the VERBOSE mode.

**@<esc>r – turns mode ‘OFF’**

STX+[OFF-R]+ETX+LRC -> When toggling out of the VERBOSE mode

15

Default: The default condition on microcontroller reset is 'ON'

**@<esc>Y – TOGGLE G4 VEND ACTIVE MODE ON/OFF.** The G4 can operate in two modes of operation. In the VEND ACTIVE ‘ON’ mode of operation the G4 provides all the MDB interface control, audit/cashless payment support, and network connectivity. In this mode a computing platform can interact with the G4 in a hybrid role to monitor a string of user text prompts (see DISPLAY PROTOCOL) as well as execute the NON-MDB-CONTROL and some MDB-CONTROL commands.

25 In the VEND ACTIVE 'OFF' mode of operation the G4 can be configured and serve as an MDB controller only. In this mode both the MDB-CONTROL and NON-MDB-CONTROL commands can be executed. While in this mode of operation the computing platform operates as a master device controlling the operation and process flow of the

system, and the G4 serves as a slave device interfacing to the vending machine and managing the control of the MDB interface. The result sting will return:

5 STX+[ON-R]+ETX+LRC -> When toggling into the VEND ACTIVE mode.

STX+[OFF-R]+ETX+LRC -> When toggling out of the VEND ACTIVE mode

10 @<esc>Y – turns mode ‘ON’  
STX+[ON-R]+ETX+LRC -> When toggling into the VEND ACTIVE mode.

@<esc>y – turns mode ‘OFF’  
STX+[OFF-R]+ETX+LRC -> When toggling out of the VEND ACTIVE mode

15 Default: The default condition on microcontroller reset is ‘ON’

@<esc>U – RETURN TO DEFAULT CONDITIONS. The G4 will return all settings to the power on/system reset default condition. The result sting will return:

20 STX+[OK-U]+ETX+LRC

20

Reset Default Conditions Include:

INTERRUPT MODE = ‘OFF’

VERBOSE MODE = ‘ON’

25 VEND ACTIVE MODE = ‘ON’

30 @<esc>P – RETURN G4 TIME AND DATE STAMP. The G4 will return the current time and date. The time and date are set by the USALIVE (host network center 808) server each time the G4 communicates with the network servers. The result sting will return:

The G4 must have the MDB state set to 'E' for ENABLED in order to find a blank record. A new record cannot be started while in a vending transaction. If a FIND BLANK RECORD command cannot execute the result string will return:

STX+[UNABLE-N]+ETX+LRC

@<esc>**B** – START A VEND SESSION. Provided the G4 VEND ACTIVE ‘ON’ mode is set, the G4 will start a vend session. The sequence to starting a vend session include:

5

G4 finds the next available blank transaction record  
 G4 loads the default data into the transaction record  
 G4 loads as the CARD DATA/ID DATA -> ‘G4-VEND’  
 G4 issues the BEGIN SESSION command to the MDB interface.

10

The result sting will return:

STX+[OK-B]+ETX+LRC

15 The G4 must have the MDB state set to ‘E’ for ENABLED in order to start a session. If a session cannot be started the result string will return:

STX+[UNABLE-B]+ETX+LRC

20 @<esc>**A**+STX+[ID DATA Up to 30 bytes]+ETX+LRC – START A DIAL-A-VEND SESSION. Provided the G4 VEND ACTIVE ‘ON’ mode is set, the G4 will start a vend session. The sequence to starting a vend session include:

25

G4 finds the next available blank transaction record  
 G4 loads the default data into the transaction record  
 G4 loads as the CARD DATA/ID DATA the ‘[DV-]+[ID DATA  
 Up to 30 bytes]’ sent as part of the command.  
 G4 issues the BEGIN SESSION command to the MDB interface.

The result sting will return:

STX+[OK-A]+ETX+LRC

5

The G4 must have the MDB state set to 'E' for ENABLED in order to start a dial-a-vend vending transaction. A new vending transaction cannot be started while in a vending transaction. If a dial-a-vend command cannot be executed the result string will return:

10

STX+[UNABLE-A]+ETX+LRC

If the LRC character does not match, or the correct ETX+LRC combination does not occur at all or in a timely fashion the result string will return:

15

STX+[NAK-A]+ETX+LRC

**@<esc>A+CARD-+[UPTO 37 BYTE TRACK 2 DATA]+ETX+LRC – START A CASHLESS TRANSACTION OR CREDIT CARD TRANSACTION.** Provided the G4 VEND ACTIVE ‘ON’ mode is set, the G4 will start a vend session. The sequence to starting a vend session include:

20

1. G4 finds the next available blank transaction record
2. G4 loads the default data into the transaction record
3. G4 loads as the CARD DATA/ID DATA the '[CREDIT CARD DATA Upto 37 bytes]' sent as part of the command.
4. Upon credit card APPROVAL the G4 issues the BEGIN SESSION command to the MDB interface.

The result sting will return:

STX+[OK-A]+ETX+LRC

5           The G4 must have the MDB state set to 'E' for ENABLED in order to start a credit card vending transaction. A new vending transaction cannot be started while in a vending transaction. If a credit card command cannot be executed the result string will return:

10           STX+[UNABLE-A]+ETX+LRC

If the LRC character does not match, or the correct ETX+LRC combination does not occur at all or in a timely fashion the result string will return:

15           STX+[NAK-A]+ETX+LRC

20           @<esc>A+SALE-+[UPTO 5 BYTES OF DATA]+ETX+LRC – VEND APPROVE response in response to the MDB TRANSACTION STRING indicating the 'R' VEND REQUEST state. The response is issued in response to the VMC issuing to the G4 the VEND REQUEST if the vend is to be approved. The 5 data byte should be in the form '00000' and should be the desired amount for the vend to charge for the vend. The VMC will report by way of the MDB TRANSACTION STRING the price of the VEND REQUESTED item. This is the value of the item as set in the vending machine. If this amount for example '00125' for \$1.25 is to be charged then the correct VEND  
25   APPROVES response should be as follows:

@<esc>A+SALE-00125+ETX+LRC

@<esc>A+VIEW-A000000C1+ETX+LRC



will return the value stored in the EEROM upper byte of address \$C1.

The G4 will return the result string and 1 byte memory value as follows:

5 STX+[OK-A-]+ ['x' 1 BYTE MEMORY VALUE] +ETX+LRC

**@<esc>A+SAVE-+['M' 1 BYTE MEMORY CODE]+['xxxxxxxx' 8 BYTES  
MEMORY LOCATION]+['x' 1 BYTE OF DATA]+ETX+LRC** – The MEMORY  
SAVE command provides a means for writing data into the G4 memory. The 1 byte  
10 MEMORY CODE selects the memory device to write the DATA. Valid MEMORY  
CODES are shown in the table above. The eight byte hexadecimal value that represents  
the MEMORY LOCATION is the physical address in memory to be written. The  
MEMORY VALUE returned in the result code in the value currently store in the  
MEMORY LOCATION – if the write was successful the value returned should be the  
15 same as the intended byte to DATA to be written.

For Example:

20 **@<esc>A+SAVE-A00000C1FF+ETX+LRC**

will write the hex value \$FF to the EEROM upper byte address location \$C1.

The G4 will return the result string and 1 byte memory value as follows:

25 STX+[OK-A-]+ ['x' 1 BYTE MEMORY VALUE] +ETX+LRC

\* note is the write was successful the BYTE MEMORY VALUE should be \$FF – the  
byte that was desired to be written to memory.

When the DEX code capture mode is switched to the 'ON' mode the following sequence of events begins:

The RAM/NOVRAM memory dedicated to the storage of DEX data is cleared.

The G4 will begin recording both the received DEX codes from the vending machine controller (VMC) and the sent DEX codes from the G4. There is RAM room for  
5 approximately 6K bytes of recorded DEX data.

When the DEX code capture mode is switched to the 'OFF' mode or automatically switches to the 'OFF' mode at the end of the DEX transfer the G4 will stop recording the DEX bus codes. A buffer dump of the DEX codes exchanged between the G4 and the  
10 VMC can be viewed by executing the @<esc>5 the DEX CAPTURED CODE BUFFER DUMP command.

Default: The default condition on microcontroller reset is 'OFF'.

15 @<esc>4 – DEX CODE CAPTURE MODE (PARSED FORMAT). The DEX controller/G4 will return the result string below toggling between 'ON' and 'OFF' of the DEX CODE CAPTURE mode. This command is for diagnostic purposes only and should not be used during normal G4 operation. The intended purpose for this command is to diagnosis DEX related transaction issues during development and or testing. The  
20 @<esc>4 command obtains DEX data in a parsed, pure format (free from all handshake and protocol exchanges (ACK, NAK, DLE, etc.). At the conclusion of the DEX data transfer the DEX CAPTURE MODE is automatically toggled 'OFF'. In most cases there will be no need to execute a second @<esc>4 command to toggle the DEX modem 'OFF'.

25

STX+[ON-I]+ETX+LRC -> When toggling into the DEX code capture mode

STX+[OFF-I]+ETX+LRC -> When toggling out of the DEX code capture mode

When the DEX code capture mode is switched to the 'ON' mode the following sequence of events begins:

- 5 The RAM/NOVRAM memory dedicated to the storage of DEX data is cleared.

The G4 will begin recording both the received DEX codes from the vending machine controller (VMC) and the sent DEX codes from the G4. There is RAM room for approximately 6K bytes of recorded DEX data.

10

When the DEX code capture mode is switched to the 'OFF' mode or automatically switches to the 'OFF' mode at the end of the DEX transfer the G4 will stop recording the DEX bus codes. A buffer dump of the DEX codes exchanged between the G4 and the VMC can be viewed by executing the @<esc>5 the DEX CAPTURED CODE BUFFER

15

DUMP command.

Default: The default condition on microcontroller reset is 'OFF'.

20

@<esc>5 – DEX CAPTURE MODE BUFFER DUMP. The DEX controller/G4 will return the result string below dumping the DEX codes passed between the G4 and the vending machine controller (VMC). The output will be formatted to indicate the codes transmitted by the VMC and the G4. This command is for diagnostic purposes only and should not be used during normal G4 operation. The intended purpose for this command is to diagnosis DEX related transaction issues during development and or testing.

25

STX+[DEX]+

-> Header

[VMC-]+VMC transmitted data

-> Data transmitted by the VMC

[G4 -]+G4 transmitted data -> Data transmitted by the G4

• • • •

ETX+LRC

5

If DEX data is obtained with the @<esc>3 command the DEX data will be parsed to remove all handshake data and VMC/G4 protocol passing. The DEX data will be pure and presented in ASCII format. If the DEX data is obtained with the @<esc>4 command the DEX data will include all the handshaking data and VMC/G4 protocol passes (ACK, NAK, DLE, etc.). The DEX data will be presented in ASCII HEX format.

**#<esc>C – ENTER SERVICE MODE ROUTINE** – The G4 will return the following result string and then enter the service mode.

15

STX+[OK-C]+ETX+LRC

**#<esc>D – SYSTEM INITIALIZATION COMMAND** – The G4 will return the following result string and then perform a system initialization sequence.

20

STX+[OK-D]+ETX+LRC

The system initialize sequence will clear memory and reload initial condition registers. The USALIVE default phone number will be loaded into memory and the G4 will be placed in the CALL HOME mode. Before correct terminal operation can be restored the G4 will require communication with the USALIVE servers.

25

#<esc>E – DISPLAY SERIAL NUMBER AND FIRMWARE VERSION – The G4 will return the following result string showing the G4 serial number and firmware version number.

5 STX+[OK-E]+ETX+LRC

Followed by the G4 serial number and firmware version number.

#<esc>F – SET CALL HOME FLAG TRUE – INITIATE CALL HOME – The G4  
10 CALL HOME flag will be set directing the terminal to begin a timer. The timer is a USALIVE setting the delays the call in process after the CALL HOME flag is set. The result code returned will be as follows:

15 STX+[OK-F]+ETX+LRC

The G4 will then in time place a call to the USALIVE servers.

#<esc>G – RETURN CURRENT STATE OF CALL HOME FLAG – The G4 will return the following result code indicating the current state of the CALL HOME flag. The states  
20 can be either TRUE or FALSE. TRUE indicating the G4 is preparing to place a call to the USALIVE servers.

STX+[TRUE-F]+ETX+LRC  
or ...  
25 STX+[FALSE-F]+ETX+LRC

#<esc>H – CLEAR THE CALL HOME FLAG – SET TO FALSE – The G4 will reset the CALL HOME flag to FALSE. The following result code will be returned.

25           **#<esc>K** – SEND CURRENT LOCAL AUTHORIZATION RECORDS – The G4 will return the result string that includes the record number, last six digits of each card, and the ‘A’ or ‘D’ suffix indicating APPROVAL or DECLINED record data as follows:



STX+[OK-M]+ETX+LRC

**#<esc>N – CLEAR APPROVAL CARD PORTION OF LOCAL AUTHORIZATION DATABASE** – The G4 terminal will clear the APPROVAL card records in the LOCAL AUTHORIZATION DATABASE and return the following result code:

5 STX+[OK-N]+ETX+LRC

#<esc>O – INITIATE A DEX QUERY MODE INQUIRY – The G4 will issue the DEX QUERY MODE string to the attached VMC. See the EVA-DTS DEX protocol standard for detail on the DEX QUERY MODE of operation. The G4 will react to the returned

10 DEX QUERY MODE result from the VMC and issue the following result string:

STX+[OK-O]+ETX+LRC

**#<esc>P – CLEAR G4 CALL-IN FLAGS** – This command clears the G4 CALL-IN flags associated with having the terminal initiate a call to USALIVE. The G4 will issue the following result string:

STX+[OK-P]+ETX+LRC

20    **#<esc>Q – VEND DECLINED** – This command is issued to instruct the G4 to decline the vend request that has been received from the vending machine VMC. See the MDB TRANSACTION STRING ‘R’ state above. The G4 will issue the following result string:

STX+[OK-Q]+ETX+LRC

25

#<esc>**R** – TOGGLE G4 VEND ASSIST MODE ON/OFF. The G4 can operate in two vending modes of operation (while the VEND ACTIVE mode is ‘OFF’). In the VEND ASSIST ‘ON’ mode of operation the G4 provides will set the ‘R’ VEND REQUEST state

flag in the MDB TRANSACTION STRING when the G4 receives the VEND REQUEST  
MDB command from the vending machine's VMC. It is then the responsibility of the  
attached computing platform to issue either the @<esc>A+SALE-[xxxxx] command to  
approve the vend and provide the sale amount, or the #<esc>Q command to DECLINE  
5 the vend. The MDB TRANSACTION STRING should be monitored to determine when  
the 'R' VEND REQUEST state is set. After the determination that the VEND REQUEST  
state is set the VEND APPROVED or VEND DECLINED command should be executed

In the VEND ASSIST 'OFF' mode of operation the G4 will automatically provide the  
10 VEND APPROVED response to the vending machine's VMC when the VEND  
REQUEST command is received from the VMC. The result sting will return:

STX+[ON-R]+ETX+LRC -> When toggling into the VEND ASSIST mode.

15 STX+[OFF-R]+ETX+LRC -> When toggling out of the VEND ASSIST mode

**#<esc>R – turns mode 'ON'**

STX+[ON-R]+ETX+LRC -> When toggling into the VEND ASSIST mode.

20 **#<esc>r – turns mode 'OFF'**

STX+[OFF-R]+ETX+LRC -> When toggling out of the VEND ASSIST mode

Default: The default condition on microcontroller reset is 'OFF'

25 **AAA - END SESSION AND PRINT RECEIPT.** A session started when the G4 is in the  
VEND ACTIVE 'ON' mode is terminated and a receipt optionally printed by send a  
string of 'AAA...'. The correct use of this command should be to send a string of at least  
six 'A' characters. Though the G4 is only looking for a combination of three consecutive  
'A's sending more is preferred.

30

**BBB – BUTTON 2 PRESS.** While in the service mode send a string of 'BBB...' has the same effect as inserting a card in the card reader to advance a menu selection or setting. The correct use of this command should be to send a string of at least three 'B' characters.

5

## G4 TEXT DISPLAY PROMPTS

### Text Display Overview:

- 10 When the G4 is in the VERBOSE 'ON' mode will send text messages out of the serial port to a display device. The display device can be the computing platform. The text messages correspond to the activity of the G4. For example, when the G4 is ready to accept cards a text prompt message of 'Please Swipe', 'A Valid Card' may be displayed.
- 15 To simplify the interface and functionality requirements of the computing platform the text prompts from the G4 can be captured and displayed on the computing platform display. Doing so alleviates the need for the computing platform to ascertain and or determine what message should be displayed to the user. In addition, allowing the G4 to manage the vending transaction, MDB interface, and text prompts removes the need for
- 20 the computing platform to get involved in the vending transaction.

The text format display protocol below illustrates how the G4 sends text prompts. The selection of the control characters is consistent with the operating functionality of many text LCD displays.

25

### Communication Interface:

The communication pins Rxd, Txd, CTS, and RTS conform to RS232 standards. A minimum of Rxd, Txd, and GND are required to implement serial communication between the G4 and a computing platform. The RTS and CTS lines only come into play from a flow control prospective when receipt data is being sent from the G4. CTS and RTS are implemented in such a way as to allow a receipt printer that has little to no

printer buffer to control the flow of data. CTS and RTS have no other purpose in non-print data communications and can be ignored or left unimplemented.

**Display and Control Codes:**

5

The G4 will send a series of display and control codes to indicate when screen initializing and format could occur. In interpreting the display and control codes the computing device will know when to blank the display area and as appropriate when and where to locate cursor positions.

10

The display and control codes are sent from the G4 as a string of hex characters. The table of display control codes is as follows:

DISPLAY CONTROL CODES	DESCRIPTION
\$FE+\$FE+\$FE	Beep Beeper
\$FD+\$FD+\$FD	Indicates print data is to follow (start print data)
\$FC+\$FC+\$FC	Indicates print data is concluded (end print data)
\$FB+\$FB+\$FB	Indicates a transaction is active – used to start a LED or set a status indicator to reflect a transaction is active. This command can be used to indicate to a user to press an ‘END’ button to end vend session.
\$FA+\$FA+\$FA	Indicates a transaction is NOT active. If a LED or status indicator is ‘ON’ resultant from the above \$FB command this command should be interrupted as negating the \$FB command.

## Table of Display Control Codes

- 5 The G4 supplies text prompts in a fixed format. The format supports two lines of text each line being a maximum of 16 characters. The format includes a leading character, which indicates the line (line 1 or 2) the text should be displayed on, up to 16 characters of text to be displayed, and a trailing character to indicate the end of the text message. When possible the text message should be formatted to contain 16 bytes. Leading spaces and trailing spaces can be used to position the text message and format the text string to 16 bytes.



25 The G4 uses a series of display codes to locate the position of the cursor. There are a maximum of 32 cursor positions – two rows of 16 characters. In addition there are cursor display codes to turn ‘ON’ a flashing cursor, turn ‘ON’ an underline cursor (show

Referring to Figure 6A there is shown a card reader and user interface system  
20 600. The card reader and user interface system 600 can be manufactured into a card

reader processor interface board 312 and as shown in Figure 3B fastened to the card reader assembly. In an exemplary embodiment system 600 is a computing platform that interconnects with system 500's interactive interface 532. In this regard the credit card and user interface 600 provide a user with user interface and display means for  
5 transacting a cashless transaction.

Interconnected with microcontroller 602 can be an input and output (I/O) interface 604. I/O interface 604 can be a plurality of data communication lines and or a plurality of communication ports such as RS232, RS485, or other similar I/O interfacing  
10 configurations. In an exemplary embodiment I/O interface 604 can be used to implement electrical interface connections to other peripheral devices. Microcontroller 602 can be any suitable microcontroller, or microprocessor. In an exemplary embodiment a microcontroller 602 can be a MICROCHIP PIC16F876-20/SP, PIC16C76-20/SP, ZILOG, MOTOROLA, UBICOM, INTEL or other similar microcontroller or  
15 microprocessor.

Interconnected with microcontroller 602 can be a display 606. Display 606 can provide message prompts and other visual information to a user. Display 606 can be any suitable display, LCD display, or flat panel display. In an exemplary embodiment a  
20 display 606 can be an OPTREX DMC-16202-NY-LY or a 16x2 line LCD character display.

A printer interface 608 can be interconnected with microcontroller 602. A printer interface 608 can be a serial communication style or Centronic style interface. In an  
25 exemplary embodiment printer interface 608 can be utilized to print receipts, coupons, and other print data.

Interconnected with microcontroller 602 can be a card reader interface 610. Card reader interface 610 can support a variety of card reader interfaces and protocols including for example and not limitation bit strobe type of card readers. Bit strobe type of card readers read predefined tracks of data from a magnetic card. To read track data the  
5 card reader can incorporate a plurality of DATA lines and DATA CLOCK lines to transfer magnetic card data. Card reader interface 610 can also support serial communications style card readers. Serial communication style card reader can incorporate TRANSMIT, RECEIVE, CLEAR TO SEND, REQUEST TO SEND control lines to transfer card data to system 500 via data communication between the interactive  
10 interfaces 532 and 614. Such magnetic card readers can include those manufactured for or by XICO, MAGTEK, NEURON, or other similar or suitable card reader.

Interconnected with microcontroller 602 can be a plurality of keypad and button inputs 612. Push button switch 308 can be electrically interconnected with button inputs  
15 612.

An interactive interface 614 can be interconnected to microcontroller 602. The interactive interface 614 operates in similar form and function to interactive interface 532.  
20

Referring to Figure 6B there is shown a card reader and user interface system 600 data communication routing switch. In an exemplary embodiment system 600 is manufactured onto the card reader and user interface board 312. Furthermore, system 500 is manufactured into VIU 100. The card reader assembly and optional printer assembly  
25 are then installed into vending equipment in such a way as to allow user access to the front faceplate 302 of the card reader assembly. Since in many cases there is little room in the vending equipment door area it is more convenient to mount the VIU 100 assembly in a different location within the vending equipment. In order to facilitate correct

operation of the card reader assembly it must be electrically connected to the VIU 100. To minimize the number of electrical connection to a single cable connected between the systems 500 and 600 a cable connection between interactive interfaces 532 and 614 can be implemented.

5

To utilize a single data communication line (transmit line and receive line) a plurality of different types of data need to be combined into a single data stream. To effectuate the combination of data into a single data stream the interactive interface communication protocol shown in the table above can be employed. To decode the data stream and route the data to its correct destination device the data communication routing switch in Figure 6B can be implemented.

10

Referring to Figure 6B there is shown an interactive interface 614 interconnected with microcontroller 602. Microcontroller 602 receives the data communication stream from the system 500's interactive interface 532 and by way of the interactive interface protocol shown in the table above decodes and routes the data to the appropriate peripheral devices. Peripheral devices shown include I/O interface 604, display 606, printer interface 608, card reader interface 610, and keypad and button inputs 612.

15

For example and not limitation print data can be packaged with the format and control codes outline in the interactive interface protocol and specification shown in the table above. Upon the data arriving at microcontroller 602, microcontroller 602 can decode that the data is print data, remove any protocol formatting characters to obtain pure print data, and then pass or forward the data to the printer interface 608. Similar processes can occur for the other peripheral devices including I/O interface 604, display 606, and card reader interface 610, and keypad and button inputs 612. Data can also be obtained from each of the peripheral devices and combined into a single data string. The

20

25

data string can be sent to the system 500 where processing can occur based in part of the data string received.

Referring to Figure 7 there is shown a transceiver and modem base unit system  
5 700 and a plurality of remote locations 804, 806, 808, 810 including a plurality of global network based data processing resources. Remote locations 804, 806, 808, and 810 can be Internet based, accessible by the Internet, or be a global network based data processing resource. In general Internet based and Internet accessible resources can be referred to as global network based data processing resources. For purposes of disclosure remote  
10 location can be referred to as remote location 804, 806, 808, and 810 and can also be referred to as global network based data processing resources.

One aspect of equipping vending equipment with a VIU 100 and or a card reader assembly and optional printer assembly is that the VIU 100 device requires a data  
15 communication connection with a plurality of remote locations. In many vending equipment locations it can be difficult to connect the VIU 100 to a physical communication line. When connecting the VIU 100 to a physical communication is difficult or undesirable the use of the transceiver and modem base unit 700 (also referred to as base unit 700) can be a more preferred data communication option. A transceiver  
20 and modem base unit 700 can be referred to as a transceiver unit 700. Transceiver unit 700 is incorporated into transceiver and modem base unit 200.

In an exemplary embodiment the transceiver unit 700 forms a wireless data link with a VIU 100 have a system 500 incorporated within. In this regard the requirement of  
25 physically connecting the VIU 100 to a communication line can be eliminated. To create a wireless data line the VIU 100 equipped with an audit-credit-interactive system 500 utilizes transceiver 524 to data communicate with transceiver unit 700's transceiver 708. Transceiver 708 is interconnected with microcontroller 702. An antenna 716 is

interconnected with transceiver 708. Antenna 716 can be of similar form and function to antenna 538. Transceiver 708 can be similar in form and function to transceiver 524.

Microcontroller 702 receives and decodes data packet information. Data packets  
5 can include command data for configuring the transceiver unit 700 and or data intended to be passed or forwarded to a plurality of remote locations by way of modem 704. Microcontroller 702 can be interconnected with modem 704. Modem 704 can be similar in form and function to modem 522.

10 Also interconnected with microcontroller 702 can be at least one of the following: a wireless interface 720, a network connection 722, an interactive interface 718, or a serial interface 724. Wireless interface 720, network connection 722, and interactive interface 718 can be referred to as a communication interface or base unit 700 communication interface. Wireless interface 720 can be similar in form and function to  
15 external modem interface 528 and or data modem 514. Network connection 722 can be similar in form and function to network interface 542. Interactive interface 718 can be similar in form and function to interactive interface 532. Serial interface 724 can be similar in form and function to interactive interface 532.

20 A plurality of remote locations can include credit bureaus such as processing bureau 804, host network centers such a host network center 808, other remote location such as remote location 806, and global network based data processing resource 810. Processing bureau 804, host network center 808, and remote location 806 can be referred to as a plurality of remote locations or remote locations. Processing bureau 804 can be a  
25 credit card processing bureau. Remote location 810 can be an Internet based data processing device or resource, or a device or resource accessible by way of the Internet – thus referred to as a global network based data processing resource.



Microcontroller 702 can be any suitable microcontroller, or microprocessor. In an exemplary embodiment a microcontroller 702 can be a MICROCHIP PIC16F876-20/SP, PIC16C76-20/SP, ZILOG, MOTOROLA, INTEL, UBICOM, or AMD.

5 Referring to Figure 8 there is shown an audit-credit-interactive system 500 interfaced to a computing platform. Figure 8 illustrates how a audit-credit-interactive system 500 can be data communication connected to a computing platform 802 by way of system 500's interactive interface 532 and computing platform 802 interactive interface. In similar form and function as the interactive interface solution between system 500 and  
10 system 600 described above, system 500 and computing platform 802 can interconnect and data communicate as described with the communication specification and protocol shown in the table above.

There can be at least two method of interconnecting a system 500's interactive  
15 interface 532 to a computing platform 802. In the first method the system 500 and the computing platform 802 can be mutually exclusive devices that share a data cable connection between the interactive interfaces. In this regard the system 500 could be manufactured separate from the computing platform 802 and later during installation interconnected together with a data cable connection between the interactive interface  
20 ports.

This method allows maximum flexibility in the selection of the computing platform 802's form and functional features as well as allowing maximum flexibility in the selection of the system 500's form and functional features.

25

A second method of interconnecting a system 500's interactive interface 532 to a computing platform 802's interactive interface can be to integrate the system 500 and computing platform 802 into a single circuit design, preferably manufactured into a

Bill acceptor 904 and coin mechanism 906 can be of a type for example and not limitation manufactured for or by MARS, COINCO, CONLUX, or other similar bill  
25 acceptor and coin mechanism type or manufacturer. Card reader 908 can be of a type for example and not limitation manufactured by or for USA TECHNOLOGIES, MARS, MARCONI, DEBITEK, SCHLUMBERGH, ACT, COINCO, EVEND, WIRCA, US WIRELESS or other similar card reader type. Online module can be of a type for

Figure 9B illustrates how an audit-credit-interactive system 500 can be configured in series with the vending machine interface 902. In this regard, the peripheral devices can be supported by the system 500's mimic MDB interface 516. The advantage off this network configuration is that the system 500 can support multiple versions and derivative versions of the NAMA MDB protocol specification. Furthermore, the system 500 can provide peripheral message emulation and message passing to effectuate the VMC's ability to data communicate to each peripheral by way of the system 500's MDB interface 518 and mimic MDB interface 516.

In an exemplary embodiment the VMC can data communicate with each peripheral device at the MDB version level of the VMC. In addition, system 500 can data communicate with the VMC and each peripheral device at the VMC version level and each peripheral MDB version or derivative version level. In this regard features supported by a peripheral device's MDB version or derivative MDB version can be utilized. In addition, data communication between the system 500 and each peripheral device effectuates the ability to remotely monitor and manage each peripheral. In this type of peripheral support system 500 servers as a data communication gateway for each peripheral device. System 500 ability to data communicate with a plurality of remote location and with each peripheral device effectuates the ability of each peripheral device to data communicate with a plurality of remote locations.

Referring to Figure 9B there is shown a system 500's MDB interface 518 interconnected with the VMC vending machine interface 902. In this relationship the system 500 can support the version of MDB protocol that the VMC firmware supports. Each of the peripheral devices including bill acceptor 904, coin mechanism 906, card reader 908, and online module 910 can then be interconnected with the system 500's mimic MDB interface 516. In this regard the system 500 can support any number of NAMA MDB protocol versions and or derivative versions of the NAMA MDB protocol.

For example and not limitation if the VMC supports NAMA MDB version 1.0 and an online module 910 supports a derivative version of the NAMA MDB protocol called advanced version 3.0 both the VMC and the online module 910 can be interconnect to a system 500 and operate correctly. In this relationship system 500 by way of MDB interface 518 and mimic MDB interface 516 data communicates with both the VMC and online module 910. The system 500 interrupts and emulates the correct device protocols as to allow the VMC to data communicate with the online module 910.

In addition, the system 500 can data communicate with the online module for the purpose of effectuating MDB command messages not supported by the VMC's MDB version. The system 500 can then selectively data communicate, to a plurality of remote locations, data related to the peripheral devices including the online module 910.

5 Furthermore, by way of system 500 the peripheral devices interconnected with the system 500's mimic MDB bus 516 can data communicate with a plurality of remote locations.

Referring to Figure 9C there is shown a audit-credit-interactive system 500 with card reader and audit functionality embodiment interfacing to a vending machine MDB  
10 bus and interfacing to a plurality of peripheral device by way of a audit-credit-interactive system 500 mimic MDB bus. In an exemplary embodiment system 500 combines the functionality of the card reader peripheral and audit or as it is commonly referred to as the online module or telemetry function thus eliminating the need for additional peripheral devices to provide these functions. In addition, the mimic MDB interface 516  
15 can optional support peripheral devices that are not compatible with the vending equipment's VMC.

Figure 9C illustrates that with a system 500 interconnected with the vending equipment interface 902 there is created two alternative bus configurations for the  
20 peripheral devices. For example and not limitation the bill acceptor 904, coin mechanism 906, as well as other types of peripheral devices can, based in part on MDB version compatibility, reside on either the vending equipment interface 902, or the system 500's mimic MDB interface 516.

25 Referring to Figure 10A and 10B there is shown a system 500 semiconductor package 1002, and an alternative system 500 module package 1002. In an exemplary embodiment a system 500 can be manufactured in to a semiconductor package. Such semiconductor packages can include industry standard through hole and surface mount

Figures 10C-10D show a system 500 integrated into semiconductor packaging 1002. Semiconductor 1002 packaging can include for example and not limitation quad flat pack style (QFP), as well as other integrated circuit industry standard package styles such as DIP, PLCC, SOP, TSOP or other suitable package style. Semiconductor package 1002 can also be a module comprising a plurality of electrical components to implement a system 500. Semiconductor package 1002 can be referred to as module 1002.



Referring to Figure 10C the system 500 shown includes microcontroller 502 interconnected with card reader interface 526, display interface 508, external peripheral interface 536, interactive interface 532, RAM/NOVRAM memory 512, timekeeper 540, flash memory 512, flash memory interface 544, RAM/NOVRAM interface 546, communication interface 548, and vending equipment interfaces 506, 516, 518, and 520. Other system 500 features can be included as may be required by the application. In addition, system 500 features shown within semiconductor package 1002 can be eliminated as may be required or desirable based on the application.

10

Timekeeper 540 can be a real time clock (RTC) for keeping track of date and time functions. Flash interface 544 can be an interface to serial and or I<sup>2</sup>C electrical erasable read only memory (EEROM), a DATA FLASH such as ATMEL DATA FLASH, serial flash memory devices, flash memory device having at least an address bus and data bus connections, or other flash memory types of devices. RAM/NOVRAM interface 546 can be a data connection to an external non-volatile read only memory device such as RAM/NOVRAM 1004. RAM/NOVRAM 1004 can be of similar form and function as RAM/NOVRAM 512.

15

20

External interconnections to semiconductor 1002 can include card reader 1012, display 606, external peripheral 1016, computing platform 802, external flash 1006, communication device 1008, interface to vending equipment 1010, and RAM/NOVRAM 1004.

25

Card reader 1012 can be an industry standard bit strobe, and serial style track 1, 2, and 3 card readers. Such card readers include for example and not limitation those manufactured for or by XICO, NEURON, and MAGTEK.



One advantage of implementing system 500 in software can be that the system 500 software solution can be implemented is mass producible generally available microcontroller devices. Such microcontroller devices can include for example and not limitation UBIKOM's line of microcontrollers, MOTOROLA, INTEL, MICROCHIP, ZILOG, and other similar or suitable microcontroller or microprocessor devices.

A second advantage of implementing system 500 in software can be that the proprietary nature of the software and its functional capabilities can more easily be concealed and protected when resident and secured within a microcontroller. In addition  
5 to the secure ability of the system 500 solutions, implementing a system 500 in a microcontroller brand or series that other design engineers are familiar with can be advantageous in easing the integration of the system 500 semiconductor package 1002 into electrical designs.

10 Further ease of integration can be achieved by implementing power converter 1020. Power converter 1020 converts input voltage obtained from the vending equipment's MDB bus via the vending machine interface 902. The output voltage from power converter can be referred to a +VCC and can power the system 500 semiconductor 1002.

15 One advantage of allowing power convert 1020 to supply power to semiconductor 1002 can be that the semiconductor can be resident on an adapter card and retrofit to existing VMC's by connection to the VMC's vending equipment interface 902. Through this single connection point the adapter card comprising the semiconductor 1002 can  
20 power itself and data communicate with the VMC by way of the VMC vending equipment interface 902.

Furthermore, the semiconductor 1002 can be integrated into the VMC controller electronics and electrically connected on the circuit board to the VMC vending  
25 equipment interface 902. Simplifying the data and power connections between semiconductor 1002 and the VMC can save time and effort in the integration of the combined VMC/system 500 solution. In addition, the fact that the system 500 can operate

Another source of VMC MDB protocol implementation variations can occur as a result of the VMC computing power and or microprocessor speed or (millions of instructions per second) MIPS capability. Microprocessor speed can influence MDB  
25 protocol implementation and message transaction speed in several ways. One such way can be in the MDB interface to the microprocessor.

The MDB protocol involves a master-slave relationship between the master vending equipment's VMC and the slave peripheral devices. In implementing the MDB protocol the master VMC initiates an MDB message command to a slave peripheral device. The slave peripheral device then has a finite amount of time to respond the VMC command message with a message response. As such the amount of time allotted for the peripheral device to respond with a MDB message response can vary from VMC to VMC. If for example and not limitation the peripheral device responds too quickly with a message response the VMC's microprocessor may not be ready and miss the return message. As a result the system 500 could fail to initialize and operate correctly. If for example and not limitation the peripheral device takes too much time to respond to the message the VMC may time-out waiting for the peripherals response message. As a result the system 500 could fail to initialize and operate correctly. An MDB MESSAGE RESPONSE timer is utilized to implement a pause from the time a MDB message is received from the VMC to the time an MDB message response from the system 500 is sent to the VMC.

The MDB initialization tuning routine 1100 determines through successive iterations of the MDB initialization sequence the optimum MDB INTER-BYTE TIME SPACING and MDB MESSAGE RESPONSE timing. Processing begins in block 1102.

5

In block 1102 the MDB INTER-BYTE INTERVAL SPACING and the MDB MESSAGE RESPONSE timers are set to a minimum range setting. Processing then moves to block 1104. Appropriate MDB MESSAGE RESPONSE time can range from a minimum range of a few microseconds to a maximum range of a several milliseconds.

10 Appropriate MDB MESSAGE RESPONSE time can range from a minimum range of less than one millisecond to a maximum range of five to ten milliseconds. Preferable a range can be from 0.5 milliseconds to 7 milliseconds and changeable by a user and or under software control.

15 In block 1104 the system 500 waits for the VMC to initiate the POLL command. In response to the POLL command the system 500 sends the JUST RESET command. Processing then moves to block 1106.

20 In block 1106 the system 500 responds to VMC MDB transaction messages with message responses in an attempt to initialize the system 500. Processing then moves to decision block 1108. Initialization of the system 500 occurs by a series of successful VMC and system 500 MDB transaction message exchanges. The system 500 can be considered successfully initialized when the VMC and the system 500 have exchanged configuration messages and the VMC has issued to the system 500 the MDB ENABLE  
25 message.

In decision block 1108 a determination is made as to whether the system 500 received the MDB ENABLE command from the VMC and if the system 500's operation

state is now enabled. If the resultant is in the affirmative that is the system 500's operation is now ENABLED then the routine is exited. If the resultant is in the negative that is the system 500's operational state is not ENABLED then processing moves to block 1110.

5

In block 1110 the MDB INTER-BYTE TIME SPACING is incrementally increased and processing moves to decision block 1112. The incrementing of the MDB INTER-BYTE TIME SPACING can be either automatic in system 500 software or manually changed by service personnel.

10

In decision block 1112 a determination is made as to whether the MDB INTER-BYTE TIME SPACING maximum range has been reached. If the resultant is in the affirmative that is the MDB INTER-BYTE TIME SPACING maximum range has been reached then processing moves to block 1114. If the resultant is in the negative that is the

15 MDB INTER-BYTE TIME SPACING maximum range has not been reached then processing returns to block 1104.

20

In block 1114 the MDB INTER-BYTE TIME SPACING is set to the initial minimum range setting. In addition, the MDB MESSAGE RESPONSE time is incremented. The incrementing of the MDB MESSAGE RESPONSE time can be either automatic in system 500 software or manually changed by service personnel. Processing then moves to decision block 1118.

25

In decision block 1118 a determination is made as to whether the MDB MESSAGE RESPONSE time maximum range has been reached. If the resultant is in the affirmative that is the maximum MDB MESSAGE RESPONSE time range has been reached then processing moves to block 1116. If the resultant is in the negative that is the



maximum MDB MESSAGE RESPONSE range has not been reached then processing moves back to block 1104.

5 In block 1116 a prompt is provided that the MDB communications between the system 500 and the VMC could not be established. The routine is then exited.

Referring to Figures 12A-12B there is shown a VIU 100 with system 500 and transceiver and modem base unit system 700 wireless protocol data communication routine 1200. In a typical application involving a VIU 100 comprising audit-credit-  
10 interactive system 500 the VIU 100 will be installed in vending equipment. In certain of those installations it may desirable to data communicate wirelessly to a transceiver system 700 instead of trying to hardwire the system 500 to a communication line. A transceiver system 700 can be referred to as a base unit or base unit 700.

15 In installations where system 500 data communicates to a plurality of remote locations by way of a wireless data connection to transceiver system 700 a protocol can be implemented to insure data integrity, security, and transceiver system 700 correct configurations. Routine 1200 can implement such a protocol between system 500 and transceiver system 700. Processing begins in block 1202.

20

In block 1202 the transceiver system 700 data communicates wirelessly an ENQ packet. The 'ENQ' packet comprises control codes that indicate the state and or current condition of the base unit 700. Base unit 700 state or condition codes include an AVAILABLE condition, BUSY condition, and a POLLING condition.

25

The AVAILABLE state indicates to any VIU 100 system 500 listening in wireless proximity to the base unit 700 that the base unit and communication interface is AVAILABLE and ready for use by any VIU 100. The communication interface includes



In block 1210 the terminal receives the ENQ and packet ID from the transceiver system 700. In accordance with system 500 data requirements the system 500 then responds as necessary with a data packet. Processing then moves to block 1212.

In block 1212 the transceiver system 700 upon receiving the data packet from the system 500 decodes the data packet. Processing then moves to decision block 1214.

5 In decision block 1214 the transceiver system 700 makes a determination as to whether the data received from the system 500 is data intended for system 700 configurations. System 700 can be referred to as the base unit or base. If the resultant is in the affirmative that is the data is configuration data for the base unit processing moves to block 1218. If the resultant is in the negative that is the data is not configuration data for the base unit then processing moves to block 1216.

10

In block 1216 the data received from the system 500 is data communicated or passed to the system 700's modem 704. Processing then moves back to block 1208.

15 In decision block 1218 a determination is made as to whether the data command received from the system 500 is a baud rate configuration command intended for modem 704 or a baud rate configuration command intended for transceiver 708.

20 In an exemplary embodiment modem 704 data communicates with microcontroller 702 at a first baud rate to effectuate data communication with a plurality of remote locations. Transceiver 708 data communicates with microcontroller 502 by way of transceiver 524 at a second baud rate to effectuate data communications between system 500 and a plurality of remote locations by way of system 700. The first baud rate and the second baud rate can be the same or different baud rated.

25 If the resultant in decision block 1218 is in the affirmative that is the data command is a baud rate configuration command then processing moves to decision block 1220. If the resultant is in the negative that is the data command received is not a baud rate configuration command then processing moves to decision block 1228.

In decision block 1230 a determination is made as to whether the received  
25 command is a modem hardware-reset command. If the resultant is in the affirmative that  
is the received data command is a modem hardware-reset command then processing  
moves to block 1232. If the resultant is in the negative that is the data command received  
is not a hardware-reset command then processing moves to block 1236.

Referring to Figure 20 there is shown a transceiver and modem base unit system 700 wireless protocol data communication routine 2000. In an exemplary embodiment the base unit 700 serves as a data gateway between the system 500 and a remote location. In operation the system 500 1) initially requests data access to the base unit 700, and if data access is granted by the base unit 700 then 2) begins utilizing the base unit 700 internal modem by transmitting and receiving data packets to and from the base unit 700. The base unit 700 manages 1) granting or denying data communication access between the system 500 terminal and the base unit 700, 2) the receiving of packetized wireless data from the system 500 for processing and data communication to the internal base unit 700 modem 704, and 3) the receiving of data from the internal base unit 700 modem, processing, packetizing, and wireless data communication via transceiver 708 to the system 500 terminal.

The data packets and wireless communication protocol can be encrypted, and error checked for data integrity. In addition, the remote hosts both credit bureau and the USALIVE network can perform their own data integrity checking. In essence the data packets can be error checked and the data received at the hosts can be error checked.

5

In an exemplary embodiment the packet by packet error checking can be turned off at the base unit 700 and system 500 and the data packets can be automatically sent multiple time. In this regard, error checking of each packet still occurs and only correct error free packets are passed to the remote location. Packets with errors are just  
10 discarded. If an error in a packet is detected it is the successive sending of the same packet that is relied upon to overcome the error. With no ACKing or NAKing on a packet by packet basis a half duplex transceiver system can send more data faster by not having to switch between send and receive modes. Since the remote location has calculated its own checksums and appended it to the data the remote location can determine if all data  
15 was received and received correctly. If the determination is made at the remote location that the data was not received correctly then the remote location can data communicate a NAK and the system will retransmit the data.

The wireless communication protocol comprises a successive try-retry algorithm  
20 to insure the wireless data transmission is received complete and error free. In the event wireless data can't be transmitted error free or packets are missing the system 500 has embedded programmable functionality to compensate for the dropped transmission.

In an exemplary embodiment, in the event the system 500 is trying to real time  
25 authorize a credit card with a remote credit bureau and the system is unable to wirelessly complete the task; instead of failing the process entirely the system 500 can terminate the wireless transmission and rely on its own internal local authorization routines 1300 or 1900 to validate the user's credit card.

In a second exemplary embodiment, if the system 500 is trying to data communicate wirelessly with a remote host location such as USALIVE and the wireless communication fails the communication can be automatically rescheduled by the system  
5 500 terminal.

Routine 2000 details the process of wireless data communication between the system 500 terminal and a remote location and or a global network based data processing resource such as global network based data processing resources 810 by way of the base  
10 unit 700.

Processing begins block 2002 where the VIU 100 listens for the 'ENQ' status packet communicated from the base unit 700. The 'ENQ' packet comprises control codes that indicate the state and or current condition of the base unit 700. Base unit 700 state or  
15 condition codes include an AVAILABLE condition, BUSY condition, and a POLLING condition.

The AVAILABLE state indicates to any VIU 100 system 500 listening in wireless proximity to the base unit 700 that the base unit and communication interface is  
20 AVAILABLE and ready for use by any VIU 100. The communication interface includes modem 704, wireless interface 720, interactive interface 718, and network connection 722. In an exemplary embodiment base unit 700 is configured to use one of the communication interface options (704, 718, 720, or 722).

25 The BUSY state indicates to any VIU 100 system 500 listening in wireless proximity to the base unit 700 that the base unit and communication interface is BUSY servicing a different VIU 100 and is unavailable for use.

The POLLING state indicates to any VIU 100 system 500 listening in wireless proximity to the base unit 700 that a remote location has requested that the VIU 100 initiate a call host sequence. This in effect triggers each VIU based on individual VIU 100 programming to initiate a call to the requesting remote location. If the current state or  
5 condition of the base unit 700 is BUSY or POLLING then the routine is exited, else processing moves to decision block 2022.

In decision block 2022 where a determination is made as to whether a system 500 terminal is requesting wireless data communication access to the base unit 700. If the  
10 resultant is in the affirmative that is there is a system 500 requesting wireless data communication access to the base unit 700 then processing moves to decision block 2004. If the resultant is in the negative that is there is no system 500 requesting data communication access to the base unit 700 then the routine is exited.

In decision block 2004 a determination is made as to whether the base unit 700 is sharing a telephone line with additional equipment, such as a fax machine. If the resultant is in the affirmative that is the base unit 700 is sharing a telephone line then processing moves to decision block 2006. If the resultant is in the negative that is the base unit 700 is not sharing a telephone line with additional equipment then processing moves to block  
15 2008.  
20

In decision block 2006 a determination is made as to whether the shared telephone line is currently in use by the additional or shared equipment. If the resultant is in the affirmative that is the telephone line is in use then processing moves to block 2010.  
25 If the resultant is in the negative that is the telephone line is available and not in use then processing moves to block 2008.



25 In block 2018 data received at the base unit's microcontroller 702 from the base unit's internal modem 704 is packetized and wirelessly data communicated via transceiver 708 to the appropriate system 500. The data received at the system 500 is validated and optionally acknowledged. A base unit 700 time-out detection and or system

500 non-acknowledge facilitates a retransmission of the data from the base unit 700. Processing then moves to decision block 2016.

5 In an exemplary embodiment data is successively and continuously handled in block 2014 and 2016 as long as data communication between the system 500 and the base unit 700 is required. The base unit 700, system 500, or the remote host can terminate data communication. Where the remote host is typically the credit card bureau, or USALIVE network.

10 In decision block 2016 a determination is made as to whether the data communication between the system 500 and the remote host by way of the base unit 700 is complete. If the resultant is in the affirmative that is the data communication is complete then processing moves to block 2020. If the resultant is in the negative that is the data communication is not complete then processing moves back to block 2014.

15 In block 2016 data communication between the G4 system, the base unit 700, and the remote host is terminated. The base unit 700 notifies all system 500s in wireless proximity to the base unit 700 that the base unit 700 is available. The system 500 concludes data communication and resumes normal terminal functionality. The routine is  
20 then exited.

Referring to Figure 13 there is shown a local transaction authorization routine 1300. A conventional card authorization through a remote processing bureau utilizing dial-up landline access to the remote processing bureau can take ten or more seconds to  
25 complete. In certain vending venue and or while vending certain type of products a ten or more second delay may be unacceptable. In these instances authorization routine 1300 can be implemented to reduce or eliminate the authorization delay while maintaining a high confidence that the card is valid. A card can be any form of ID including a credit

card, magnetic card, wireless phone, a personal digital assistant PDA, private label card, smart card, hotel room card, radio frequency RFID identification, biometric, and or other similar or suitable form of ID. Processing begins in decision block 1302.

5           In decision block 1302 a determination is made as to whether the LOCAL  
AUTHORIZATION FLAG is set for this pass. In an exemplary embodiment system 500  
can be programmed to locally authorize a card based in part on an iterative process,  
which allows for the local authorization routine to be invoked, at a minimum, on the first  
pass and subsequently at any successive pass up to the last pass. The current pass through  
10 the routine is referred to as the CURRENT AUTHORIZATION ATTEMPT. The last  
pass is predetermined and is referred to as the MAXIMUM AUTHORIZATION  
ATTEMPTS LIMIT. The LOCAL AUTHORIZATION FLAG determines on which  
iterative pass the local authorization routine will be invoked. The iterative pass in which  
the LOCAL AUTHORIZATION FLAG will be set and the local authorization routine  
15 invoked is referred to as the LOCAL AUTHORIZATION ROUTINE ENTRY  
COUNTER.

          In a first example and not limitation the local authorization can be invoked on the  
first pass. In this case no remote location will be contacted unless the local authorization  
20 results in a declined card response. In a second example the local authorization flag may  
be set for the second pass. In this case the system 500 will first try to remotely authorize  
the card. If the remote processing bureau is unavailable or unable to authorize the card  
then on the second pass the local authorization routine will be invoked.

25           If the resultant in decision block 1302 is in the affirmative that is the LOCAL  
AUTHORIZATION flag is set then processing moves to decision block 1304. If the  
resultant is in the negative that is the LOCAL AUTHORIZATION flag is not set then  
processing moves to block 1308.

In block 1308 system 500 initiates a data communication for the purpose of authorizing the current card with the processing bureau 804. Processing moves to block 1312.

In block 1312 a local database within system 500 can be updated. This local data can include positive cards, which are cards that have previously been successfully approved. In addition, this local database can include negative cards, which are cards that  
5 have previously been declined. Processing then moves to decision block 1314.

In decision block 1310 the card is tested for its appearance in the system 500 local databases. If the resultant is in the affirmative that is the card does not appear in a negative database and or the card appears in the positive database then processing moves  
10 to block 1312. If the resultant is in the negative that is the card appears in the negative database and or does not appear in the positive database then processing moves to block 1308.

In decision block 1314 a determination is made as to whether the card has been  
15 approved. If the resultant is in the affirmative that is the card has been approved then processing moves to block 1316. If the resultant is in the negative that is the card has been declined than processing moves to decision block 1318.

In decision block 1318 a determination is made as to whether the MAXIMUM  
20 AUTHORIZATION ATTEMPTS LIMIT has been reached. The MAXIMUM AUTHORIZATION ATTEMPTS LIMIT is the count of the number of iterative authorization passes through routine 1300. If the resultant is in the affirmative that is the MAXIMUM AUTHORIZATION ATTEMPTS LIMIT has not been reached then processing moves back to decision block 1302. If the resultant is in the negative that is  
25 the MAXIMUM AUTHORIZATION ATTEMPTS LIMIT has been reached then the routine is exited the card is reported as declined.

In block 1316 the transaction is reported as authorized and the vending equipment is enabled for operation. The routine is then exited.

Referring to Figures 19A-19B there is shown a local authorization routine 1900.

5 Routine 1900 is an exemplary embodiment of a local authorization routine. Routine 1900 utilize locally stored databases to selectively approve and decline a card transaction. Similar to routine 1300, routine 1900 can be implemented to reduce or eliminate the authorization delay while maintaining a high confidence that the card is valid. A card can be any form of ID including a credit card, magnetic card, wireless phone, a personal  
10 digital assistant PDA, private label card, smart card, hotel room card, radio frequency RFID identification, biometric, and or other similar or suitable form of ID. Processing begins in decision block 1902.

In decision block 1902 a test is performed to determine if the card presented is  
15 expired. If the resultant is in the affirmative that is the card is expired then processing moves to block 1914. If the resultant is in the negative that is the card is not expired then processing moves to decision block 1904.

In decision block 1904 a test is performed to determine if the modulo-10 of the  
20 card presented is correct. If the resultant is in the affirmative that is the modulo-10 check digit match then processing move to decision block 1906. If the result is in the negative that is the modulo-10 check digit does not match then processing moves to block 1914.

In block 1914 the transaction authorization is declined and the routine is exited.  
25

In decision block 1906 a test is made to determine in the MAXIMUM APPROVAL RESET HOUR has been reached. If the resultant is in the affirmative that is the MAXIMUM RESET HOUR has been reached then processing moves to block 1910.



25 In decision block 1908 a determination is made as to whether the LOCAL AUTHORIZATION FLAG is set for this pass. In an exemplary embodiment system 500 can be programmed to locally authorize a card based in part on an iterative process, which allows for the local authorization routine to be invoked, at a minimum, on the first pass and subsequently at any successive pass up to the last pass. The current pass through



USA-010-03

- 117 -

PATENT

the routine is referred to as the CURRENT AUTHORIZATION ATTEMPT. The last pass is predetermined and is referred to as the MAXIMUM AUTHORIZATION ATTEMPTS LIMIT. The LOCAL AUTHORIZATION FLAG determines on which iterative pass the local authorization routine will be invoked. The iterative pass in which  
 5 the LOCAL AUTHORIZATION FLAG will be set and the local authorization routine invoked is referred to as the LOCAL AUTHORIZATION ROUTINE ENTRY COUNTER.

In a first example the local authorization can be invoked on the first pass. In this  
 10 case no remote location will be contacted unless the local authorization results in a declined card response. In a second example the local authorization flag may be set for the second pass. In this case the system 500 will first try to remotely authorize the card. If the remote processing bureau is unavailable or unable to authorize the card then on the second pass the local authorization routine will be invoked.

15

If the resultant in decision block 1908 is in the affirmative that is the LOCAL AUTHORIZATION flag is set then processing moves to decision block 1912. If the resultant is in the negative that is the LOCAL AUTHORIZATION flag is not set then processing moves to block 1926.

20

In decision block 1912 a test is performed to determine is the card presented for authorization is currently in the local DECLINED database. Is the resultant is in the affirmative that is the card is in the DECLINED database then processing moves to block 1926. If the resultant is in the negative that is the card is not in the DECLINED database  
 25 then processing moves to block 1916.

In block 1916 the APPROVED database is queried to determine how many occurrences of the current card are already in the database (having previously been

USA-010-03

- 118 -

PATENT

presented and locally authorized). This is referred to as the NUMBER OF CARD OCCURRENCE IN APPROVED DATABASE. Processing then moves to block 1920.

In block 1920 a comparison of the NUMBER OF CARD OCCURRENCES IN APPROVED DATABASE is made to the MAXIMUM OCCURRENCES WARNING LIMIT. Processing then moves to decision block 1918.

The MAXIMUM OCCURRENCES WARNING LIMIT is a remotely programmable preset variable that indicates the number of repeat occurrences the same card number can be presented for local authorization before a remote authorization is forced. In an exemplary embodiment, it may be desirable to accept for local authorization a card no more the 2 time before that card will be authorized at a remote location. The presumption is that only the remote location has access to real time card data validity. In that case a card may be locally authorized that is in fact not a valid active card. To minimize the risk of authorizing a bad card the MAXIMUM OCCURRENCES WARNING LIMIT can be set as desired to limit exposure on a bad card. For example and not limitation the MAXIMUM OCCURRENCE WARNING LIMIT can be set to two. In this case the card will be accepted and locally authorized twice before a remote authorization step is forced.

20

In decision block 1918 a test is performed to determine if the NUMBER OF CARD OCCURRENCES IN APPROVED DATABASE exceed the MAXIMUM OCCURRENCE WARNING LIMIT. If the resultant is in the affirmative that is the NUMBER OF CARD OCCURRENCES IN APPROVED DATABASE exceed the MAXIMUM OCCURRENCE WARNING LIMIT then processing moves to block 1926. If the resultant is in the negative that is the NUMBER OF CARD OCCURRENCES IN APPROVED DATABASE does not exceed the MAXIMUM OCCURRENCE WARNING LIMIT then processing moves to block 1922.

25

In block 1922 the card data is added to the APPROVED database and the transaction is approved. Processing moves to block 1924.

5 In block 1924 the transaction is authorized and the user is allowed to utilize the vending equipment. The routine is then exited.

In block 1926 system 500 initiates a remote data communication for the purpose of authorizing the current card with the processing bureau 804. Processing moves to  
10 decision block 1930.

In decision block 1930 a test is performed to determine if during the remote authorization process a communication line failure was detected. Such a failure could include for example and not limitation a phone line being busy or unavailable, or a server  
15 being unavailable. If the resultant is in the affirmative that is a communication line failure was detected then processing moves to block 1928. If the resultant is in the negative that is a communication line failure was not detected then processing moves to decision block 1934.

20 In block 1928 the APPROVED database is queried to determine how many occurrences of the current card are already in the database (having previously been presented and locally authorized). This is referred to as the NUMBER OF CARD OCCURRENCE IN APPROVED DATABASE. Processing then moves to block 1932.

25 In block 1932 a comparison of the NUMBER OF CARD OCCURRENCES IN APPROVED DATABASE is made to the MAXIMUM OCCURRENCES STOP LIMIT. Processing then moves to decision block 1918.

The MAXIMUM OCCURRENCES STOP LIMIT is a remotely programmable preset variable that indicates the number of times the same card number can be presented for local authorization before a remote authorization is forced. In an exemplary embodiment, it may be desirable to accept for local authorization a card no more than  
5 three times before that card will be authorized at a remote location.

The difference between the MAXIMUM OCCURRENCE WARNING LIMIT and the MAXIMUM OCCURRENCE STOP LIMIT is that when the WARNING limit is exceeded a remote authorization will be attempted. If the remote authorization attempt  
10 fails due to communication related issues then the STOP limit will serve as a determinant in approving or denying the current transaction. If the STOP limit is not exceeded the local authorization processing will continue. If the STOP limit is exceeded the card will be declined. This feature in affect minimizes the possibility of inadvertently declining a card based on a communication line failure.  
15

In decision block 1940 a test is performed to determine if the NUMBER OF CARD OCCURRENCES IN APPROVED DATABASE exceed the MAXIMUM OCCURRENCE STOP LIMIT. If the resultant is in the affirmative that is the NUMBER OF CARD OCCURRENCES IN APPROVED DATABASE exceed the MAXIMUM  
20 OCCURRENCE STOP LIMIT then processing moves to block 1942. If the resultant is in the negative that is the NUMBER OF CARD OCCURRENCES IN APPROVED DATABASE does not exceed the MAXIMUM OCCURRENCE STOP LIMIT then processing moves to block 1944.

25 In block 1944 the card data is added to the APPROVED database and the transaction is approved. Processing moves to block 1946.

In block 1946 the transaction is authorized and the user is allowed to utilize the vending equipment. The routine is then exited.

5 In decision block 1934 a test is made to determine if the card was approved. If the resultant is in the affirmative that is the card was approved then processing moves to block 1938. If the resultant is in the negative that is the card was declined then processing moves to block 1936.

10 In block 1936 the card is entered into the DECLINED database. If multiple occurrences of the card appear in the DECLINED database the duplicate occurrences are removed. Processing then moves to block 1942.

15 In block 1938 all occurrences of the card data are removed from the APPROVED database. The transaction is approved and processing moves to block 1946.

Referring to Figure 14 there is shown an international transaction authorization and settlement routine 1400. Standard transaction processing fees for low cost sales can be significant. International card processing can incur even more transaction processing fees in the form of currency conversion fees. Currency conversion fees are fees incurred  
20 when currency is converted from one countries currency to another. To minimize the standard transaction processing fees and to minimize and or eliminate the currency conversion fees routine 1400 can be implemented. Processing begins in block 1402.

25 In block 1402 the local authorization routine 1300 is executed. Processing moves to decision block 1404.

In decision block 1404 a determination is made as to whether a remote data communication to a processing bureau 804 is required. If the resultant is in the

affirmative that is the local authorization is approved and a remote authorization is not required then processing moves to block 1408. If the resultant is in the negative that is the local authorization was declined or failed and a remote data communication with processing bureau 804 is required then processing moves to block 1406.

5

In block 1406 authorization through a network connection to a remote host network 808 and or processing bureau 804 is executed. Processing then moves to decision block 1418.

10

In decision block 1418 a determination is made as to whether the remote authorization was approved. If the resultant is in the affirmative that is the remote authorization was approved then processing moves to block 1408. If the resultant is in the negative that is the remote authorization failed or was declined then the card is declined and the routine is exited.

15

In block 1408 the vending equipment is enabled and vending can occur. Processing then moves to block 1410.

20

In block 1410 a batch of locally authorized transactions is data communicated to a remote location (the remote location being another country) by way of a network connection. In this regard locally authorized transactions can be moved from the country in which the vending sale occurred to the country where the transactions will be processed with a processing center. Processing then moves to block 1412. The transfer of locally authorized transactions can occur at a predetermined time including hourly, daily, weekly, monthly, or other desirable time interval.

25

In block 1412 the server receiving the locally authorized transactions from a plurality of remotely located system 500 authorizes and settles each locally authorized

transaction. The process of settlement effectuates the transfer of funds from the cardholder to the merchant. In this regard, the transaction is authorized and settled in the same country currency avoiding any currency conversion fees. Additionally, since transactions can be aggregated from a plurality of system 500 in a plurality of countries the transaction and currency volumes increase. These increases in transaction volumes coupled with efficient batching of transactions to the processing bureau can result in the lowest possible standard transaction processing fees. Processing then moves to block 1414.

10 In block 1414 the funds generated from the authorization and settlement of the locally authorized transaction can be electronically transferred back to a bank in the country in which the vending sale occurred, or the country of choice. Such a transfer can be accomplished by an electronic funds transfer (EFT), or other similar or desirable method for transfer funds. The transfer of funds can occur at a predetermined time including hourly, daily, weekly, monthly, or other desirable time interval. Processing then moves to block 1416.

20 In block 1416 reporting requirements can be effectuated as required and electronically transmitted to the appropriate parties. Such a reporting cycle can be referred to as a remittance cycle and can be utilized by all parties having involvement in the transactions to among other things verify fund transfers, and monitor vending equipment operational efficiencies. The remittance cycle can occur at a predetermined time including hourly, daily, weekly, monthly, or other desirable time interval. The routine is then exited.

25

Referring to Figure 15 there is shown a data communication transaction message parsing routine 1500. In an exemplary embodiment system 500 can generate data and transactions relating to vending equipment DEX data, vending equipment MDB data,



vend transaction data, financial transaction data, system 500 diagnostic data, and other types of data and transactions. While a system 500 has data communication access to a remote host network center 808 the system 500 can data communicate the mixed batch or varying types and kinds of data to the host network center 808 servers. It is at the host  
5 network centers 808 that the data and transaction must be parsed and handled in different methods. Such parsing method can include forwarding data to a subsequent server, storing data in a database, data processing to produce a new result and then acting on the resultant data, storing and forwarding transaction data including card transaction data for authorization and settlement, as well as implementing other methods for handle mixed  
10 batch data parsing. Processing begins in block 1502

In block 1502 a host data connection is initiated and established between the system 500 and the host network center 808. Such a data connection can be a dial-up connection, and Internet based connection, or other suitable data connections. Processing  
15 then moves to block 1504.

In block 1504 the system 500 terminal configuration data is exchanged between the system 500 and the host network servers. This terminal configuration data effectuates the ability to remotely manage the terminal operational parameters including the  
20 terminals firmware version form a remote host network center 808. Processing then moves to block 1506.

In block 1506 the host network server receives a data stream from the system 500. The data stream can comprise a mixed batch of operational data, marketing data,  
25 transaction data, and other types of data. Processing then moves to block 1508.

In block 1606 a MAXIMUM VEND ITEM LIMIT is determined and set. The MAXIMUM VEND ITEM LIMIT is the maximum number of items that can be vended on a single authorization. The MAXIMUM VEND ITEM LIMIT can be stored as part of the system 500's terminal configuration file and remotely managed by way of the remote host network center 808. In an exemplary embodiment the MAXIMUM VEND ITEM LIMIT can range from one to ten items. Processing then moves to block 1610.

25 In decision block 1622 a determination is made as to whether the NO ACTIVITY  
TIMER LIMIT has been reached. If the resultant is in the affirmative that is the NO  
ACTIVITY TIMER LIMIT has not reached the limit then processing moves to decision

block 1624. If the resultant is in the negative that is the NO ACTIVITY TIME LIMIT has been reached then processing moves to block 1626.

5 In block 1626 the end session sequence is started. The end session sequence includes waiting for the vending equipment to complete any last vends, ending the vending session, saving sales record data, optionally printing a receipt, and any other end sequence steps that may be required. The routine is then exited.

10 In decision block 1624 a determination is made as to whether the user has pressed the end transaction button. If the resultant is in the affirmative that is the user has pressed the end transaction button then processing moves to block 1626. If the resultant is in the negative that is the user has not pressed the end transaction button the processing moves to decision block 1628.

15 In decision block 1628 a determination is made as to whether a VEND REQUEST MDB command has been received from the vending equipment's VMC. If the resultant is in the affirmative that is the VEND REQUEST has been received then processing moves to block 1630. If the resultant is in the negative that is the VEND REQUEST command was not received then processing moves back to decision block  
20 1622.

In block 1630 the VEND REQUEST command is processed and a VEND APPROVED or VEND DENIED response message is data communicated from the system 500 to the requesting VMC. Processing moves to decision block 1632.

25

In decision block 1632 a determination is made as to whether the MAXIMUM VEND ITEM LIMIT has been reached. If the resultant is in the affirmative that is the MAXIMUM VEND ITEM LIMIT has been reached then processing moves back to block

1626. If the resultant is in the negative that is the MAXIMUM VEND ITEM LIMIT has not been reached then processing moves to decision block 1634.

5 In decision block 1634 a determination is made as to whether the AUTHORIZED VALUE LIMIT has been reached. If the resultant is in the affirmative that is the AUTHORIZED VALUE LIMIT has been reached then processing moves back to block 1626. If the resultant is in the negative that is the AUTHORIZED VALUE LIMIT has not been reached then processing moves to decision block 1638.

10 In decision block 1638 a determination is made as to whether the user has pressed the end transaction button. If the resultant is in the affirmative that is the user has pressed the end transaction button the processing moves back to block 1626. If the resultant is in the negative that is the user has not pressed the end transaction button then processing moves to block 1636.

15 In block 1636 the RE-VEND TIMER is reset to zero. Processing moves to block 1640.

20 In block 1640 a vending session is started. A vending session is started by sending the BEGIN SESSION MDB command to the vending equipment's VMC. Processing moves to decision block 1642.

25 In decision block 1642 a determination is made as to whether the RE-VEND TIMER has reached the RE-VEND TIMER LIMIT. If the resultant is in the affirmative that is the RE-VEND TIMER has reached the RE-VEND TIMER LIMIT the processing moves back to block 1626. If the resultant is in the negative that is the RE-VEND TIMER has not reached the RE-VEND TIMER LIMIT then processing moves to decision block 1646.

In decision block 1646 a determination is made as to whether the user has pressed the end transaction button. If the resultant is in the affirmative that is the user has pressed the end transaction button the processing moves back to block 1626. If the resultant is in the negative that is the user has not pressed the end transaction button then processing moves to block 1644.

In decision block 1644 a determination is made as to whether a VEND REQUEST MDB command has been received from the vending equipment's VMC. If the resultant is in the affirmative that is the VEND REQUEST has been received then processing moves to block 1648. If the resultant is in the negative that is the VEND REQUEST command was not received then processing moves back to decision block 1642.

In block 1648 the VEND REQUEST command is processed and a VEND APPROVED or VEND DENIED response message is data communicated from the system 500 to the requesting VMC. Processing moves to back to decision block 1632.

Referring to Figure 17 there is shown a data communication sweeping, processing, and data forwarding routine 1700. In an exemplary embodiment the host network center 808 accumulates a plurality of different kinds of parsed data transaction in a temporary data structure. Such a parsing and temporary data structure can be implemented as disclosed in routine 1500. To move the data transactions from the temporary data structure a more permanent data structure and or host network sever routine 1700 can be implemented. Processing begins in block 1702.

In block 1702 the transaction store in the temporary data structure are swept into an operational database. Such an operational database can be implemented as a SQL

database, ORACLE database, flat file database, DB2 database, and or a combination of different kinds and types of databases. Processing then moves to block 1704.

5 In block 1704 locally authorized transactions that have not been previously authorized are authorized with a processing bureau 804. This authorization after the vending sale has occurred can be referred to as post authorization. Processing then moves to block 1706.

10 In block 1706 any transactions including the previously post authorized transaction are settled with the processing bureau 804. The process of settlement effectuates the transfer of funds from the cardholder to the merchant. Settlement after the vending sale has occurred can be referred to as post settlement or post settle. Processing then moves to block 1708.

15 In block 1708 any refund transactions generated by the host network center customer service are processed. Refund transactions can occur when a previously settled transaction requires some portion of the sale amount be refunded to the cardholder. Customer service can generate a refund transaction by querying from an operation database the original transaction and then initiating a refund transaction based in part on  
20 the queried customer's original transaction. Processing then moves to block 1710.

In block 1710 data related to vending equipment DEX and MDB detail can be converted as required and data communicated to databases, and or other servers. The process of converting the DEX and MDB data can involve parsing and repackaging the  
25 data into a desired data warehousing interface format. Alternatively, the DEX and MDB data can be posted to a server where customers can by way of a network connection to the host network center 808 download the data.



In addition to the convert and forward functionality the data handled can be measured and counted as desired for the purpose of billing for the service of gather data from a remote system 500 and delivering the data to a customer's desired location. Measurement and counting can include for example and not limitation measuring file and  
5 or data size, measuring the frequency the data is gathered, counting the number of times data is gathered and or forwarded, measuring access to the host network center 808, or by other suitable measurement and counting methods and or criteria. Processing moves to block 1712.

10 In block 1712 the funds collected from the processing of transaction can remitted to the customer as required by EFT or other desirable method. The funds remitted can have service fees deducted from them such that their EFT amount is less than the total processed transaction amount. In this regard customer will not have to be billed for services. The deducting of service fees from the flow of funds can eliminate the need to  
15 invoice a customer for service. The routine is then exited.

Referring to Figure 18A-18B there is shown a mimic MDB interface port routine 1800. In an exemplary embodiment the system 500 can serve as a MDB protocol conversion gateway. In this regard the system 500 can emulate and interpolate VMC  
20 MDB messages for a plurality of peripheral device. In addition, the system 500 can act as a MDB master or MDB slave device allowing the system 500 to support peripheral devices the VMC cannot. Routine 1800 implements the system 500 functionality to support the MDB interface 518 and the mimic MDB interface 516. Processing begins in block 1802.

25

In block 1802 the VMC and system 500 exchange MDB message commands by way of the VMC vending equipment interface 902 and the system 500's MDB interface

25 In decision block 1812 a determination is made as to whether the MDB command message is a bill acceptor command message. If the resultant is in the affirmative that is the MDB command message is a bill acceptor MDB command message then processing moves to block 1814. If the resultant is in the negative that is then MDB command

25 In block 1822 the system 500 optionally sends as a master device a MDB command message to the peripherals interconnected with the mimic MDB interface 516. Such peripherals can include coin mechanism, bill acceptor or validator, or other peripherals on the mimic MDB interface 516. If the system 500 does not have a MDB

command message to send processing moves back to block 1802. If the system 500 has a command message to send the command message is sent to the desired peripheral device and processing moves to block 1824.

- 5           In block 1824 the system 500 receives any device response messages resultant from the sent MDB message. Processing then moves to decision block 1828.

- 10           In decision block 1828 a determination is made as to whether any received message or data on the mimic MDB bus needs to be forwarded or passed to the VMC by way of the MDB interface 518. If the resultant is in the affirmative that is the system has a command message or data to send to the VMC processing moves to block 1830. If the resultant is in the negative that is the system 500 does not has a command message or data to send to the VMC processing moves to block 1826.

- 15           In block 1826 the terminal system 500 can manage the data received from the peripheral device as required. Processing moves back block 1802.

- 20           In block 1830 the terminal system 500 responds to the VMC POLL command message and responds by passing the command message and or data from the peripheral device to the VMC. Processing moves back to block 1802.

- 25           While this invention has been described with reference to specific embodiments, it is not necessarily limited thereto. Accordingly, the appended claims should be construed to encompass not only those forms and embodiments of the invention specifically described above, but to such other forms and embodiments, as may be devised by those skilled in the art without departing from its true spirit and scope.